HAAGA-HELIA
University of Applied Sciences

Software Product Business: Business Models and Processes
A Practical Approach to Licensing Principles and a Software Asset Management

Iina Haatainen

Thesis

Business Information Technology

2008

HAAGA-HELIA
University of Applied Sciences

Abstract

03.11.2008

Business Information Technology

Authors Iina Haatainen	**Group** BITE
Software Product Business: Business Models and Processes A Practical Approach to Licensing Principles and a Software Asset Management	**Number of pages and appendices** 65 + 1
Supervisors Tuusa Tenho	

The purpose of this thesis is to introduce the software product business, business models and processes and give a practical approach to licensing principles and a software asset management. The software product business is living a new era due to break-through innovations concerning application, workstation and server virtualization. This process has influenced on the whole branch and forced manufacturers, distributors and resellers to their licensing policies and earning logics. The ICT-cluster is in front of the new challenges, because of the shortened life cycle of the products and continuously moving economical aspects caused by the customers, which are not willing to pay according to same licensing principles as before.

The software asset management is both the challenge and the answer for the customer organizations to get direct and indirect software licensing savings and better license management. The traditional licensing policy and management are the challenges for the customers and the changes mentioned above will make it more difficult than ever. The need of the overall understanding is to acquire dedicated persons or specialized service providers to plan and manage the software asset management processes to get the best possible features and benefits of the software asset management concept.

The added-value of this thesis is to give a set of tools to understand the complex licensing policy and pay attention to the possibilities of the software asset management concept.

Key words

Software Asset Management, Software License Agreements, License Management, Software Product Business

Table of Contents

FIGURES

TABLES

1 Introduction

The software asset management is a challenging area, because the software product business is developing very fast, even too fast, so that the available books and study materials are much behind than a contemporary user needs and the access to the latest innovations and materials are very limited. A practical approach has played very important role, because the sources for this thesis has been achieved mostly via the employer courtesy as widely as the confidential material including standardization protocols, for example the use of ISO 19770-1 software asset management is very strictly controlled.

1.1 Background to the Research

The software product licensing is on the edge, because of the rapid innovations and technical development on ICT - sector. It was extremely difficult to find out up-to-date material for daily work and functioning processes for the software asset management. There clearly existed the urgent need to collect all available information into a one place and develop it for own and related job positions to increase the learning time and make faster the overall understanding on licensing environment and the software asset management. The software product business has been and still a quite closed society and the branch educates and trains their employees self, but usually using insufficient or non-existing study material. The need for every new employee is to collect and create own set of usable tools and material.

1.2 Purpose and Objectives of the Research

The purpose of this thesis has been to introduce and give a view of software product business objectives in Finland and relate it to common licensing principles and a software asset management. Besides of spending huge amount of money on the licensing and software related projects the future offers great opportunities to create more effective software asset management. The material is collected for easing the management and understanding the software licensing and related concepts.

At first rose up to question what is the software product business environment and the actors working in it today in Finland. The second issue was to find out the common principles and legal aspects of licensing and finally to clarify the software asset management scope, objectives, processes and overall concept with factors influencing on its structure?

1.3 Scope of the Research

The scope of this research is limited to a practical approach to get a better understanding of the terminology and processes of the software product business and its special features in Finland by the view of a reseller organization. The research investigates the life cycle and services of the ready software product more than the development of the product. The software asset management (SAM) as a service is mentioned only to clarify the concept and also a very important open source will need more specified own research.

2 Software Business

The Software Business is a fascinating business sector, perhaps its fascination is based on attractive market values, in Europe worth of 76 milliards and globally 207 milliards Euros. The market value growth in the year 2006 in Europe was 6, 6 percent and the growth of Finland's Software Business revenue was world's fastest. (Laurila, J. Tietoviikko. 2007)

The growth has slowed down since and it is certain, that the financial crises started from the United States on spring 2007, will have an impact on the Software Business. Finland's Software sector has been health with help of new, tightly the industry trends following innovations. Although, it's exactly measurement has been challenging, because the Software Business has been examined as a part of the medal industry from the beginning (Hyvönen 2003. 3).

The Software Business cluster belongs to Information and Communication Technologies, referred as ICT. The Software Business can be limited as the business where trading concentrates on software programs and the related services (Hyvönen 2003.1).

2.1 The Software Business Categories

The Software Business as a concept covers three overlapping business categories based on the traded objects and the productization level (Lamberg, T. 2008). The categories are software products, customer tailored software and embedded software (Hyvönen 2003. 3). The research concentrates more closely on the software products and the related services in the next chapters. At this point is essential to gain an overall understanding of the Software business environments. The services in the Software Business can be third party's labor with fixed pricing, consulting or value-added services embedded in the actual product's pricing.

When speaking of the Software industry, it includes software product and customer tailored business, but Software product business Industry contains only the software products (see the Figure 1.). Sometimes the Software Business has been used misleadingly only for Software industry (Hyvönen 2003. 3). The companies operating on the industrial sector offer software product and software designing and implementing services. In addition to this, Software Business covers also distributing, product selling and selling support and other related services to the software products. (Nukari, Saukkonen & Seppänen 2003. 162)

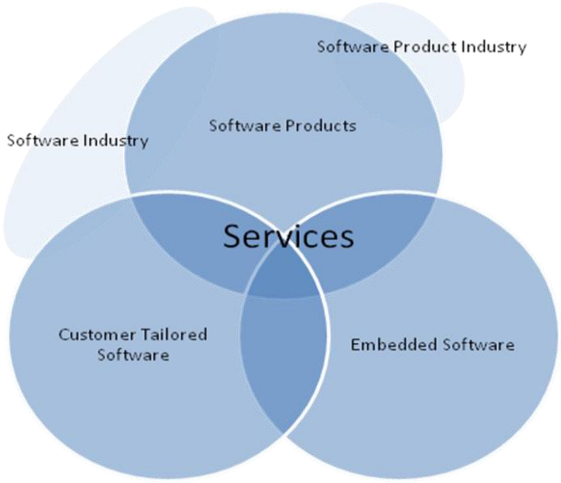


Figure 1: The Software business categories (Hyvönen 2003. 7)

2.1.1 Software Products

High level of the productization, small production costs, big volumes and wide customer segments are the keys for success when dealing with the software products. Lamberg (2008) separates the software products as single products, not a piece of any other product, delivered to end-customers without any customized tailoring. These identical software products are usually called as packed, mass-market or shrink-wrap software. In addition, the software product business includes installation, training, support and maybe even some customization restricted on the software programs' interfaces, but the core of trading is the software product. (Lamberg, T. www.swbusiness.fi. 2008) The well-known examples of the software products are Microsoft Office applications for PCs and laptops, like Word and Excel, or Office suites, like Office Standard and Office Professional Plus. See the more accurate classification on Figure 2.

SOFTWARE CLASSIFICATION (Based on NAPCS)

- Software Publishing
  - SYSTEM SOFTWARE
  - Operating Systems Software
  - Network Software
    - Network management software
    - Server software
    - Security and encryption software
    - Middleware
    - Other Network Software
  - Database Management Software
  - Development Tools and Programming Languages Software
    - Software testing tools and testing software
    - Program development tools
    - Programming languages software
    - Other Development Tools and Programming Languages Software
  - Other System Software
  - APPLICATION SOFTWARE
  - General Business Productivity Applications
    - Office suite applications
    - Word processors
    - Spreadsheets
    - Simple databases
    - Graphics applications
    - Project management software
    - Computer-based training software
    - Other General Business Productivity
  - Home Use Applications
    - Games
    - Reference
    - Home education
    - Other Home Use Applications
  - Cross-Industry Application Software
    - Professional accounting software
    - Human resource management software
    - Customer relations management software
    - Geographic Information System software
    - Web page/site design software
    - Other Cross-Industry Application Software
  - Vertical Market Application Software
  - Utilities Software
    - Compression programs
    - Antivirus software
    - Search engines
    - Font
    - File viewers
    - Voice recognition software
    - Other Utilities Software
  - Other Application Software

Figure 2: Software Product classification (Tyrväinen, Lamberg, Nukari, Saukkonen, Seppänen & Warsta. 2005. 42)

The required productization level for software products sets many challenges when thinking of needed innovations to the product development, funding, risk management and (international) marketing (Hyvönen 2003. 2). The tempo of new versions releases is fast, because of the growing competition and a single product's life cycle is much shorter than it used to be in beginning of 90's. The software products without updates can be security risks. A battle between software vendors and hackers spends time, money and resources.

2.1.2 Customer Tailored Software

Tailored software is customized individually for every customer. The solution cannot be duplicated even in a standard situation (Hyvönen 2003. 3). The software is developed to meet specific business requirements and it can be embedded into customer's current IT-environment or it can be a stand-alone system.

The customized software mainstreams are developing the stand-alone systems from a scratch or the system, which integrates or is an integration tool between other software components. The implementation is usually done in projects with carefully project management. The projects can be short-term or last years. The processes of customer tailored software are typically divided on defining, designing, programming and testing phases (Haikala & Märijärvi 2003. 35).

The customer tailored software can be divided on

System Software and Software tools, which are designed to optimize computer's application program operations and general applications like word and spreadsheet processing.

Engineering and scientific software, which are used for modeling, for example, natural phenomenon.

Knowledge-based and expert systems, which are collection systems for example some certain industry's expertise.

Business software, which is are designed to automate office routines or produce information for monitoring and guiding a company's operations.

Process control system and process automation systems, which monitor production processes.
(Haikala & Märijärvi 2003. 17)

2.1.3 Embedded Software

Embedded software has common features with the software products and customized software. It can be identical software with wide distribution areas or it can be tailored according customer's requirements. The conceptual thing is that the embedded software is a software component in some other product. It cannot be distributed without the whole product and the business is focused on the actual product entity, like a mobile phone. The mobile phones have identical operation systems as embedded software. The other example could be factory's automation system. (Hyvönen 2003. 3) Altogether, the embedded software is a critical part of the bigger picture and it cannot be separated as stand-alone product. The business revenues are counted based on the other factors, for example in numbers of the hardware pieces.

When the Software Business is discussed in generally, it usually does not count the embedded software despite of the common classification.

2.2 The Common Features of the Software Business Models

The business models are used to describe features and rules producing more value to the software business. The models illustrate profits or other profitable benefits for software manufacturing companies or business units related to software business. The models help to recognize and identify the actors and their relations more accurately on the software business cluster. The business models interact with each other on different levels and together with external forces such as business strategy, competition environment, customer needs, financial needs and resourcing. The core elements of the business models are categorized as the product strategy, the model of the service and the implementation, the distribution model and the earning logic. The earning logic is the most crucial element combining all the other models as daily processes for producing a business actor's actual revenue. (Rajala, Rossi & Tuunainen 2003. 8 - 11)

When discussing the business and the product strategy, the strategy is certain processes for business actors to survive and succeed. It can be seen as a dynamic puzzle, where essential is to examine the over-all picture, which is continuously living. The business models must change internally, following the external pressures, which in ICT cluster are closely related to the innovation level. (Kostamo 1999. 22)

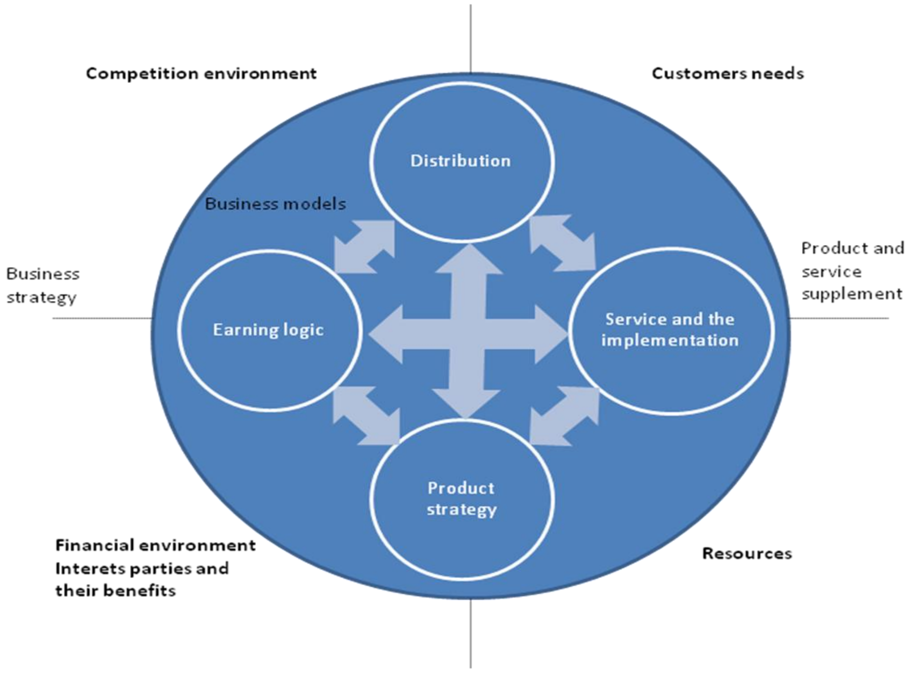


Figure 3: The elements of the business models (Rajala, Rossi & Tuunainen. 2003. 11)

The product strategy is the strategy for selling a core product and its core features. How our software is better than a competitor's (Kostamo 1999. 22)? The strategy defines the level of research and development. How soon a new update or a new release should be published?
The business model answers to the questions "*What* we are doing?" and "*How* the product is implemented?". The answer of How-question draws the outlines of the product's development model (Rajala, Rossi & Tuunainen 2003. 11).

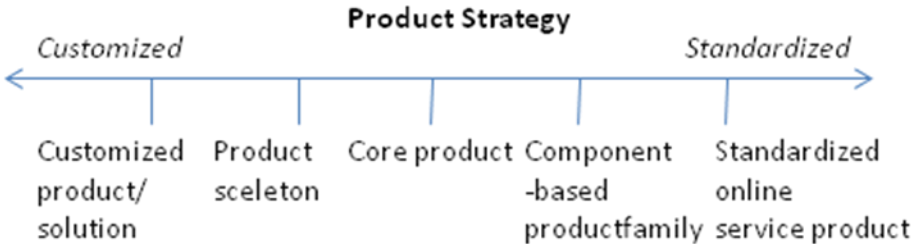


Figure 4: The focus examples to product research and development (Rajala, Rossi & Tuunainen 2003. 11)

Usually a young software company's focus is on customized products or solutions. When the company has managed to reach a solid ground for its business, it tries to move the focus on more standardized product for higher manufacture level. (Nukari & co 2003. 166) In Finland a typical feature is that starting companies use open source, discussed more later on, when establishing the product development. This removes many restrictions set by other interest parties, but it also limits the financial resources and support.

The service and implementation model describes how the core product is delivered to end-customers as a well-functioning entity. Rajala & co (2003) introduce terms the product's refinement and service providing level as a part of it. In practice the refinement and service providing level describe highly developed productization, which has become more and more popular in the software business. One reason can be a good profit marginal without any further additional costs and usually the service agreements are done for long time period or automatically as a continuing service. The human resources reserved for providing services are not counted in. (Rajala & co 2003. 12)

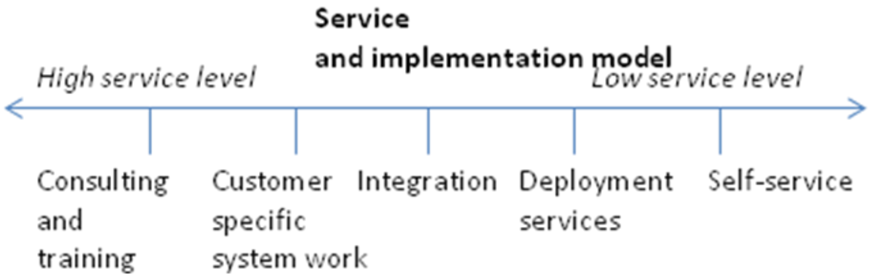


Figure 5: The service and implementation model. (Rajala & co 2003. 12)

The typical feature to high-level services is that the total value is difficult to estimate in beforehand, but the potential is high. The more high-level service and implementation are, the more experienced resources and processes are needed. When discussing on productizing the consulting services, the core product can be seen as material and immaterial parts related directly to a benefit, what customer gets when buying a service. (Sipilä 1996. 62)

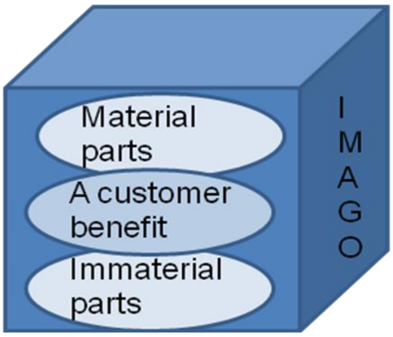


Figure 6: The core product's customer benefit. (Sipilä 1996. 63)

The distribution model describes channel or channels of marketing, buying and paying the core product. Often the most profitable situation is when there is more than one distributor with different customer segments. The better is, if a relationship between a customer and a distributor is close. This ensures the channel over-all functionality. The fundamental decisions concerning the distribution model are for example, how the big picture of marketing and distributing of the core product and core service is organized and who really are the resellers and marketers. The model also describes the selling decisions, for example what kind of an agreement finalized the selling process. Will the delivered software or possible service or both be tailored for customer's specific needs or will the end-result be identical for every one? The service and implementation model is concretely related to the distribution model, when defining who, from distribution channel, implements core product's or service's possible customization. (Rajala & co 2003. 12 - 13)

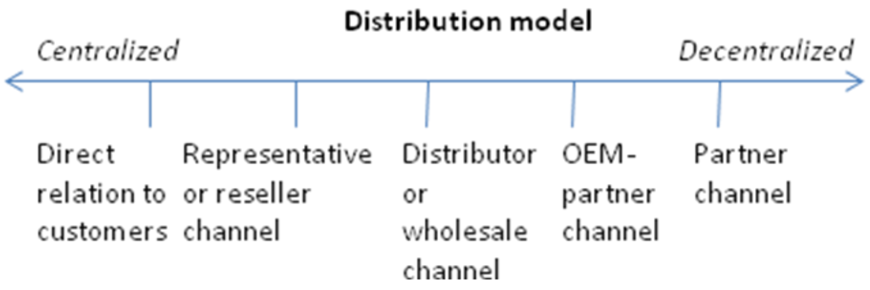

Distribution model
Centralized
Decentralized
Direct relation to customers
Representative or reseller channel
Distributor or wholesale channel
OEM-partner channel
Partner channel

Figure 7: The Distribution model. (Rajala & co 2003. 13)

Usually in Finland the software business distribution channels are built through representative or reseller and distributor and wholesale channels. The common situation is organizations of the software product vendors' are too light for totally centralized model and the lack of resources is replaced using the channels. The smaller Finnish software manufacturing companies are using more centralized model, because of the customized software or solution approach, introduced before.

The earning logic is the most essential model and through it the business is channelized to be profitable or unprofitable. This model determines, among other cost-income decisions, the pricing of the core product or solution.

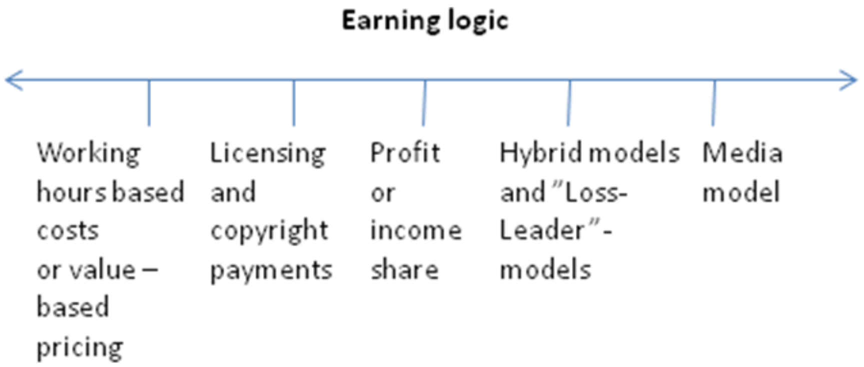

Earning logic
Working hours based costs or value – based pricing
Licensing and copyright payments
Profit or income share
Hybrid models and "Loss-Leader"-models
Media model

Figure 8: The earning logic examples. (Rajala & co 2003. 14)

"The media model", according Rajala & co (2003) in the earning logic Figure 8 describes an example, how the money is received from a third party for advertisement. "The Loss-Leader" model in turn, describes sacrificing the profit for the future's revenue, for example selling the core product in negative price and compensating the loss selling the related services. (Rajala & co 2003. 14)

The environmental factors (see the Figure 3) restrict the company's core product and services operations. The competition environment and - positioning strongly shape the software business markets effecting concretely on the profit margins. Today's software life cycle has become fast based on high-speed innovations and though this all the time evolving customer needs. The product development and research is expensive and need continuous financing. When discussing on the software business resourcing the most relevant resources are professionalism and potency. The innovation must be classified also as the resource. It is an engine of the whole software industry, as well as the whole ICT-cluster. (Rajala & co 2003. 11)

3 The Software Product Business

The software product business is based on the software products. The provided customer-related services and deployment services have often an essential role in the software product business, but not dominating. The core business object is always the software product. (Lamberg 2003. 152) The software product business according to the product strategy is more emphasized on as the standardized product. The products usually have wide distribution channels with help of several different sale networks.

Table 1: The software product business areas, when the sold product includes a software product copy and a license (Tyrväinen, Lamberg, Nukari, Saukkonen, Seppänen & Warsta. 2005. 4)

Software product state to end-customer	**Customer does not pay for the software product license**	**Customer pays also for the software product license**	**Customer pays only for the software product license**
Software embedded in the service	Service (implemented or supported using the software)	Service + license (service is supported using the software)	Sofware as a service (Service implemented using the software)
Stand-alone software	Open source, freeware, shareware	Software product + services (configuration, installation, training…)	Shrink and wrap-software product
Software embedded in the hardware	Hardware, system (software product with hardware)	Hardware + embedded software product (separately license fee for a software product)	

The delivered product to end-users is an immaterial use right. A customer buy or lease the right to use the software product according to vendor's set software agreement and product use rights. Violating these terms can cause a punishment and the end-user is obligated to reimburse to a person or a company owning the software copyrights.

Software product business can be seen as started in the United States during the late 1960's. The first move was done by the undeniable market leader International Business Machines Corporation, later referred as IBM. IBM decided to sell separately the software applications and the hardware. In the year 1969 this was a fully new approach. Välimäki (2006) underlines that the software business analytics tend to think this, at least, as a symbolic starting year of the software product business, which was necessary for the fast growth of the software business in 1970s. Mikko Välimäki (2006) mentions that according to Campbell-Kelly (2003) the reasons behind IBM's decision were more versatile computer use, growing software development costs, lack of professional programmers and IBM's own, almost as a standard, became platform. (Välimäki 2006. 137)

3.1 Software Product Business in Finland

In Finland, software product business started during 1970's with help of few domestic software products. It gained more attention during 1990's, when several development projects, were created for supporting the software product companies. (Lamberg 2003. 152) The growth since has been intensive and the speed hasn´t slowed down not until the recent years (Laurila, J. Tietoviikko. 2007).

The most famous and successful project for software product companies in Finland was the SPIN project (Software Product INdustry) during the years 2000 - 2003 by the Finnish funding agency for Technology and Innovation, later referred as TEKES. The value of SPIN was approximately 70 million Euros. The project's objectives were to help Finnish software product companies to internationalize. It also covered the other interest groups, companies and actors in the supporting infrastructure. The SPIN covered all together 110 company and 14 research projects. The end-results supported successfully Finland's software product business product and productization on the key sectors, networking and co-operation, developing high-quality and competitive expertise, operational environment and support services and internationalization. (TEKES 2003. SPIN - software products 2000 - 2003)

The creator of the SPIN-project, TEKES, is one of Finland's biggest sponsors in software business cluster. It helps and built the cluster's business together with the software companies. The agency also actively participates and supports related research work and studies in universities, polytechnics and other institutes. TEKES is vital for Finnish software business and it has funded many small companies when starting the business. The agency has customer companies yearly approximately 3000 and 50 different educational institutes. TEKES starts yearly over 2000 research and development project, which is a remarkable number (TEKES 2008). In addition to TEKES, there exist several other organizations also like regional Technology centers, The Finnish Information Processing Association (FIPA) and The Finnish

Association of Technology industries for developing and supporting Finnish software product business.

TEKES, together with other Software product cluster's actors have created the cluster vision until the year 2015. The vision's objectives are to lift the software product business revenue to 15 milliards Euros, cluster employment level to 60 000 employees and to have 50 companies listed in the United States stock, NASDAQ (Hietanen & Nurmi 2005. 1 -2). This challenging vision for the software industry has not so far come true. The growth numbers have not been as high as expected and many traditional medal industry companies have shown more potential numbers. The amount of domestic software product business companies is approximately 1000 and the most of the companies have only a couple employees. Of course, it is needed to remember that software product as a business sector employee covers all the companies related to the software product business. (Lukkari, J. Tekniikka&Talous. 2008)

3.1.1 The Finnish Software Product Business Characteristics

The characteristics of Finnish software product business cluster are based on the results of the national Software Industry Survey 2007 conducted by Helsinki University of Technology. The survey was the tenth annual Software industry survey and its missions are to create a better perspective of the software product business, underlining the business growth and internationalization. The Survey's scope was companies, who owned the offered licenses or related services or any other tightly business linked services. The study was implemented as the mail and web based survey and it was sent to 2616 companies representing the software product industry in general. Due the small response rate, the survey's results can be seen only suggestive. For this research point of view, the survey gives to a reader overall understanding needed for the study objectives, when mirroring the software product business environment in Finland and the supply chain introduced on the next chapter. (Rönkkö, Eloranta, Mustaniemi, Mutanen & Kontio 2007. 2 -5)

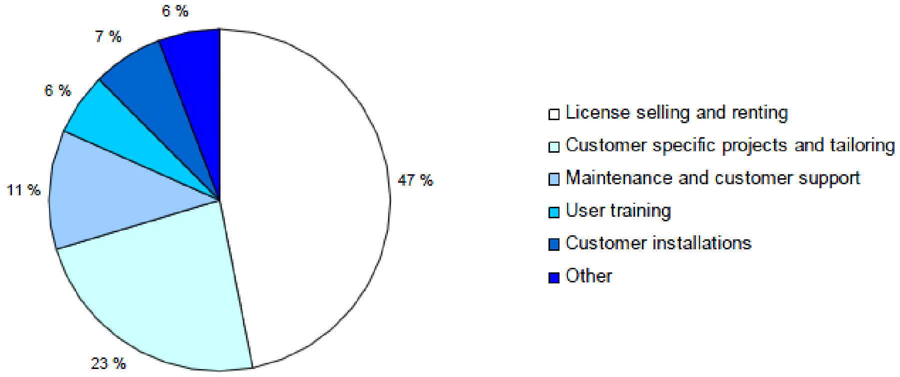


Figure 9: Composition of a typical Customer Sales Revenue of the Main Product n=250 (Rönkkö & co 2007. 23)

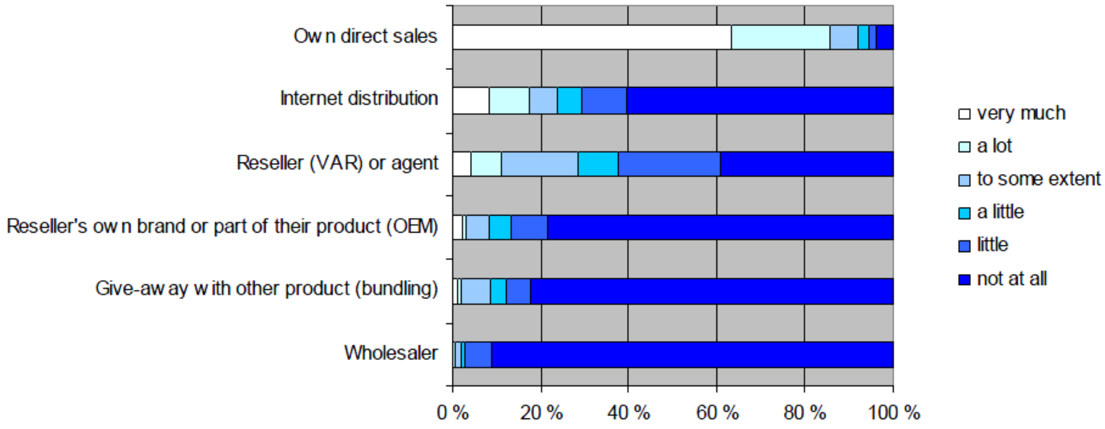
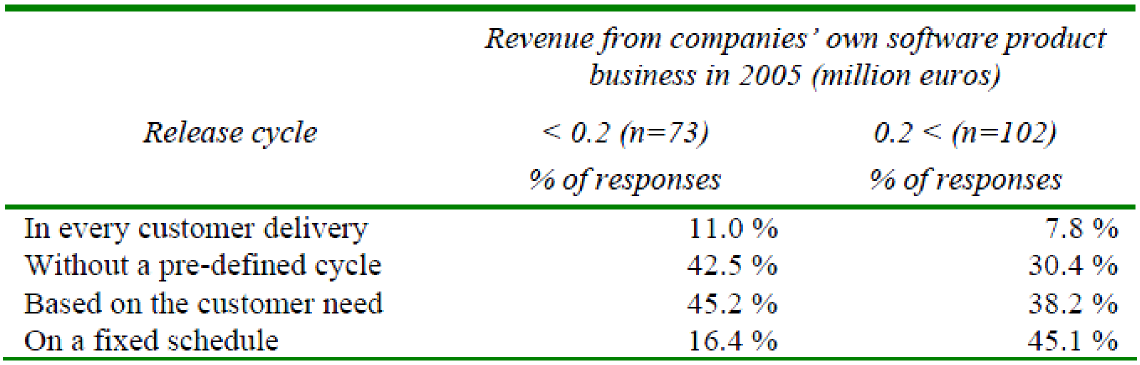


Figure 10: The different Sales Channels and Their Usage n=257 (Rönkkö & co 2007. 24)

Table 2: End-Users in the Different Market Segments (Rönkkö & co 2007. 25)

End users	*Revenue from companies' own software product business in 2006 (million euros)*				
End-user	*< 0.2*	*0.2-0.99*	*1-1.99*	*2-2.99*	*3 <*
Micro enterprise	16%	10%	0%	8%	8%
Small enterprise	45%	40%	39%	50%	25%
Medium enterprise	47%	55%	39%	42%	38%
Large enterprise	40%	57%	39%	83%	75%
Public sector	25%	35%	44%	33%	42%
Private consumer	12%	7%	0%	17%	8%
Number of cases	102	88	18	12	24

Table 3: Basis of Releasing a New Version of the Main Product (Rönkkö & co 2007. 25)



	Revenue from companies' own software product business in 2005 (million euros)	
Release cycle	*< 0.2 (n=73)*	*0.2 < (n=102)*
	% of responses	*% of responses*
In every customer delivery	11.0 %	7.8 %
Without a pre-defined cycle	42.5 %	30.4 %
Based on the customer need	45.2 %	38.2 %
On a fixed schedule	16.4 %	45.1 %

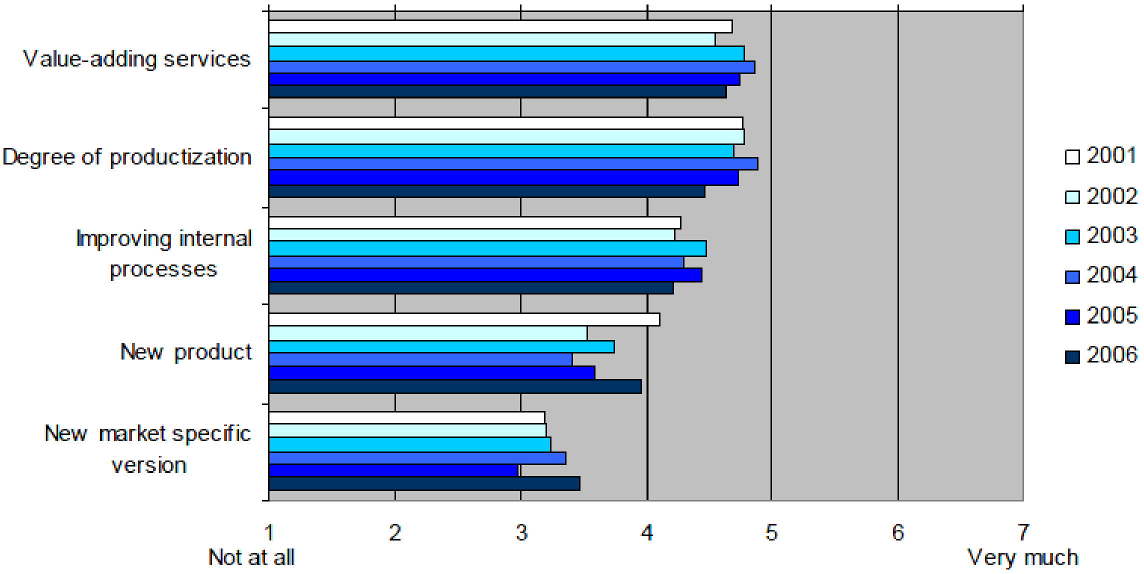


Figure 11: Product Development Emphasis 2001 - 2006 n=98-102 (Rönkkö & co 2007. 27)

The software product business in Finland is strongly concentrated on selling and renting the licenses (Figure 9) and the use of different distributor channels is rare (Figure 10). This can emphasize that the companies participated to the survey are not the on the beginning in a supply chain, discussed later on, or the productization level is not so far, that the distributor channels would be crucial for companies' earning logic. The popular customer segments are middle-sized or bigger companies (Table 2).

3.2 The General Features of the Software Product Business Supply Chain

The supply chain of the software product business can be classified roughly in three different actors and in six different end-customers (see the Figure 12). The actors can have several interest groups between each other. The networking is essential under the environmental business pressures like competition, customer needs, finance and resources, introduced on the previous chapter.

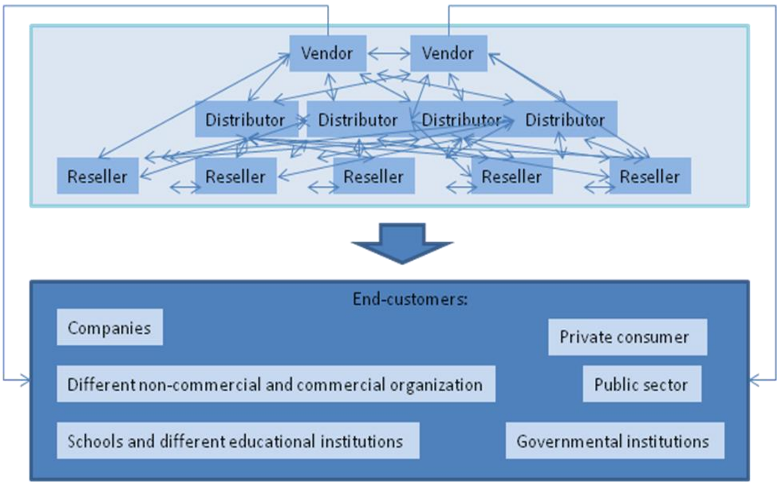


Figure 12: The supply chain of the software product business

Vendor or the product manufacturer owns the software product copyright and possible some other legal privileges, like a patent and a trademark. These are discussed more accurately on the chapter 4. The vendor can have a very centralized distribution channel, where the right of use the software product is sold directly to the end-customer. This option is more popular among small domestic software product manufacturers, where the productization level is low. The more common is that the vendor's uses more decentralized distribution model, like tight networks between distributors, resellers or partners. The partners can also be the end-customers, who get the products in a lower price in the name of the partnership, for example the end-customer can be seen as solution partner to the vendor increasing this way the overall product profit. One reason for decentralized approach is usually the vendor's own very light selling and marketing organization. The actual resources are normally focused on the research and development operations and maintaining the already existing product portfolio.

The most beneficial situation to the vendor is when it has wide, functioning and professional distributor channels. These kinds of channels familiarize the technology and improve its use among the end-customers the way, which satisfies all the interest groups. The most used tactic to ensure the quality of professionalism of the distributors, resellers and partners is to provide different kinds of free or charged marketing and technical certifications. According to these certificates, the distributors, resellers or partners can be qualified to participate in vendor's programs, for example Microsoft's gold certified partner-program. The programs can have several levels and motivators usually are the lower purchase prices on the higher program level or possible free software use for own test -or production environment.

The normal option to motivate the next supply levels can also be different commissions and fees paid according to the certain percentage from an individual product sale or according to overall sold product volume, for example during the vendor's annual year or quarter. Some vendors have also so called forecast-systems. A reseller who forecasted a sale first, gets a percentage based reward, nevertheless who made the actual sale. These types of rewarding systems have an effect on the pricing can cost a certain type Leader-Loss pricing among the distributors and resellers. There are several rewarding alternatives, which can be vendor - or vendor's product related. In the last hand, the question is how every interest group is managed to gain some profit from the sale.

Vendor sells the software products, or more accurate, the licenses to the distributors, resellers or directly to the end-customers. The distributors sell licenses to the resellers or directly to the end-customers with their own pricing and risk. Usually the risks are minimized with fast purchase processes so that the license is bough from the vendor after the actual sale. The distributor is in a supply chain responsible for product logistics and warehousing. It has an essential operational role the supply chain actors.

The distributor can offer different software consulting and concrete help with finalizing the sale. The importance of this kind of value-added service has grown remarkable during the last years, especially for the smaller resellers, who do not have enough, for example, technical sale resources. This development has forced some distributors to think their earning strategy again and reform as value-added distributors, later referred as VAD. VAD as a term is for distributing carefully chosen software products, usually couple vendor's main product palette instead of the traditional multi-variety. VAD policy could be culminated on the common phrase "less is more". Value-added distributors sell or offer freely, in addition to the sold product license, the related product consulting, technical supporting and marketing and selling resources or professionalism. (Mäntylä, J. Tietoviikko. 2007) The different distribution channel approaches offered by VADs have become popular. It has given more possibilities to resellers to sell software products with a wider variety than before and concentrate on their core business, for example the services.

The reseller is a conceptual actor operating between the end-customer and the other software product interest groups. Resellers purchase the licenses from the distributor or directly from the vendor in their own risk, but usually after the actual sale. The reseller can be seen as the supply chain's last layer providing essential information about end-customer needs to the vendor for further development. The resellers also share the new product and license information released by the vendor to the end-customers. In the customer's point of view the reseller is an independent actor offering all the competing technologies.

Value-added resellers, later referred as VARs, use the same technique than VADs. The reseller has limited the represented software products and concentrates more on offering productized value-added services, for example licensing consulting, training, deploying and configuring and maintaining the software. The tight competition has forced the actual software product margins low so Loss-Leader models have become more and more profitable business, when resellers try to distinguish from the other market actors.

The end-customers can be classified in six main roles according to their operational status and based on the general license agreement programs, discussed more later on. These segments are not related to the marketing segments defined by individual companies operating on the software business sector. The customer roles are companies, different non-commercial or commercial organizations, schools and different educational institutes, public sector, governmental institution and private consumers (see the Figure 12).

The companies are entities, which have economical profit making - activities, nevertheless the company's legal form. A company can be owned by a private single person or interest group or many private persons and interest groups. Some educational institutes, public sector or the government can be among the mentioned interest groups, but the ownership percenttage or other conditions are not fulfilling vendors' set restrictions for non-company license programs. The customer can also be non-commercial and commercial organizations, for example different innovation associations, like TEKES or humanitarian organizations. The profit margins are very limited if none.

Schools and other organizational institutes are the customers without any profits and the entity's operations can be seen providing an educational or sophistical value. Public sector customers are cities and municipalities. The governmental institutions are the customers, who are qualified using the services of Finnish government's central procurement unit, Hansel Ltd.

Hansel Ltd is fully owned by Finnish government and it takes care for tendering the products and services needed on the public administration. (Hansel Oy) A private consumer is a single person, who buys the software for the private use.

3.3 The Future Aspects of the Software Product Business

The Software product business has rapidly grown during the past decades and forecasts have been promising. Now the speed has slowed down and the necessary, long-desired global expansion is still waiting to happen. The internationalization, ensuring professional labor and removing other possible barriers for the growth are the biggest challenges today. (Laurila, J. Tietoviikko. 2007) Hietanen & co (2005) summarize that the software product business trends based on the wide literal analyses and according to the software product business specialists are the development of the information and network societies, fast expansion of the technological innovations, globalization and tightening competition (Hietanen & Nurmi. 2005. 16)

In practical these trends are constantly growing virtualization and more powerful machines with multicore processors. The virtualization unchains software from the hardware-based limitations and for example a one physical machine can run several virtual instances with heavy operational processes. This renews the alternatives how the software is delivered and where the software actually runs. The local installations are not obligatory anymore and this is the problem for traditional software product licensing. (Robinson, B. Infoworld.com. 2006)

The software product business has to adapt and create new business concepts, which overcome the physical limitations. According to Pentikäinen Juho (Tietoviikko. 2007) Gardner has forecast the software as a service (SaaS) -type application services produced by a third party will be more popular. SaaS is discussed more closely later on. The customer will buy, instead of perpetual licenses, according to the actual software product usage for example per user per monthly basis. For a customer this kind of evolution decreases the software and hardware investments (Tietoviikko. 2008). New systems and applications can run even on a web browser served through The Internet, so called cloud computing (a service from the Internet "cloud"), without the further deployment, monitoring or maintaining by the customer.

4 The Juridical Foundation of the Software Product Business

When trading is concentrating on the software products, the laws and regulations create the foundation to the whole industry. The most essential laws in the software product business are the act of copyright protection and the patent act. These are also referred as intellectual property (IP).

Table 4: The immaterial rights used for computer programs (Välimäki 2006. 129)

LEVEL	ACTS	IMPACT
code	copyright act, maybe patent act, trade secret	copying a code in any shape, can impact on different interfaces
interfaces	maybe patent act, trade secret	using commercially interface data, can impact on the code
architecture	copyright act, maybe trade secret	copying an architecture, no impacts on any other levels
user interface	copyright act, registered design, maybe trademark act	imitating a user interface, no impact on any other levels
functionality	patent act, utility models	copying a functionality, can impact on other levels

The history of the software's juridical status can be seen begun in the United Stated of America. The commission, set for solving problems related to the software copyright protection, decided to support the regular copyright law in the year 1978. According to this decision the USA became the first country in 1980, where the software products were separately taken legally in notice. The European countries slowly followed this practice during the 80's. WIPO (World Intellectual Property Organization) announced finally in 1985, that "a great number of participants developed arguments in favor of recognizing copyright protection of computer programs; patentability of computer programs *per se* had been ruled out under the virtually every country… copyright, in its development, had proved to be flexible enough to extend to works of technical nature". Soon after this the European Union standardized the practice creating the directive 91/250/EEC, which was finally approved in the year 1991. (Välimäki 2006. 9 - 12) After this, the legislation continued living, but the foundation created on the 80's stayed as the standard practice for the software creation protection.

4.1 The Copyright Protection of the Software

The Finland's copyright protection act defines that, the entity, which has created any literal or artistic work, owns the creation's copyright. The creation's nature can be fictional or any other literal work, oral performance, composition, theatrical, cinematographical, photographical or any else artistic work. The work can also be architectural, an artistic craft work or a product of the industrial art. There is no matter how the work is expressed. (Finland's copyright protection Act 8.7.1961/404 1§). The new addition, related to the software, was made in year 1991 according the European Union's directive; the literal work can also be a map, other drawing, plastically designed work or **software product** (40 b § 11.1.1991/34). In this perspective, the software programs are interpreted as a literal work. This has caused conflicts, because the nature of the software program cannot be so directly pointed out than in the traditional literal work. The creativity shown in programming is more versatile and the intelligence or its innovation value is harder to measure. (Pitkänen 2003. 78)

Mikko Välimäki (2006) introduces the copyright act as a very carefully limited monopoly right, which main aim is to seek a solution for the production of information communication technology. If there is no any kind of protection for the creators of the innovative work, the software production volumes would decrease. In the other hand, also too tight restrictions would decrease remarkably the volumes. In practice, the copyright act protects this kind of situations, but still underlines that the created work is not the creator's property. It is only protected for a certain time limit, defined in beforehand. The copyright protection lasts for 70 years. The noticeable here is that the timing starts after the last programmer of the copyrighted program is dead. The same policy applies on every entity protected by the copyright act. However, the evolution of the software is so fast that, in the other words, the copyright protection lasts forever. The business value for copying or distributing the software program after approximately 100 years is nonexistent. (Välimäki 2006. 13 - 17)

The copyright protection is good enough when thinking it from the software product business point of view. It is generated automatically when a program is programmed without any additional registrations and without any interest party's privileges. The copyrighted software cannot be copied, distributed or redesigned without permission. More controversial innovation scenarios are algorithms, principles, structural logic of the user interfaces and the actual use of the software. Those can be seen more as basic human values and cannot so forth be limited with the copyright. (Välimäki 2006. 13 - 17)

The owner of the copyrighted software cannot be a computer. The owner can only be a programmer, who has written the source code. When discussing on programming as a business resource, the copyright is needed to separately transform as an asset of the business entity. The act itself defines two exceptions, which are a collaboration work and a collection work. In the collaboration work there are no possibilities to distinguish creators. When two or more authors have created a mutual work, they can together or individually claim an ownership preventing a possible malpractice. (Finland's copyright act 8.7.1961/404 6§) The collection work is a work wholly or partially collected from other works. The collector of the entity owns the collection's copyright, but is not able to restrict the use of the original works. (Finland's copyright Act 8.7.1961/404 5§)

When a software program is reflected as the collaboration work, its licensing requires every programmer's acceptance. In practice, the software cannot be licensed without a mutual decision. However, an author can give the rights to the third party without a permission of the other programmers. A good example of the collaboration work is free Linux operation system. Linux consists of many components programmed by numerous programmers. Linux Torvalds embedded artificially these components together. A commercial computer game can be seen also as the collection work. It ties together game's programmer, graphic artist and music composer. (Välimäki 2006. 30)

When observing the copyright protection through the software product business, the copyright is transferred to a company without additional agreement when programmer works for the employer. If the programmer is not an employee of the company, the copyright must be transferred separately. (Välimäki 2006. 29 - 32)

Other than the software programs, databases are protected by a list protection. The list protection ensures that all data gathered using significant resources or time into a single database cannot be unauthorized copier or shown public. The database protection lasts for 15 years and timing is started when database is not modified anymore or it is published as a finished data resource. (Pitkänen 2003. 82)

4.2 Patenting Software

The Software can be patented like any other innovation. In Finland patens are applied from national board of patens and registration of Finland, later referred as NBPR. A term patent stands for "a right to exclude. The holder of a patent has the right to exclude others from commercial utilization of the invention in accordance with his or her patent. Forms of commercial utilization include such acts as making, selling, using and importing of a patented product" and patents are used for "protecting the results of the research and development work of an enterprise or an individual inventor. … The right to exclude covers a restricted territory; it is in force in the countries where patent has been applied for and granted. The right to exclude is in force for a limited period of time, generally no longer than for 20 years from the filing date of the application. For a patent to remain in force, its holder must pay annual maintenance fees, so called renewal fees, for it. … A patent may be sold or licensed. In compensation the licensee pays the patent holder for instance a specified percentage of the income produced by the invention (royalty)" (NBPR. 2006).

Noticeable is, that the patent must be applied in every country separately, where a software product patent is desired for protection. The patenting challenges are country-specific, but sometimes the amount of needed capital is remarkable and the bureaucracy can be so overwhelming, that it is recommended to hire specialized agency for filling the application forms. This probably is the biggest reason why patenting the software is seen unnecessary and the copyright protection is argument to be enough. (Välimäki 2006. 89)

The patents have raised many discussions and strong opinions. The ultimate fear is that the patent will turn out to be too efficient way to protect technical innovations like software programs. Instead of protecting, it might restrict and limit the development processes of the new ideas. The software patents can also be problematic for software standardization point of view slowing it down and frustrating end-users with compatibility problems between the software formats. The patent is argument as favoring monopolies and forgetting small open source companies. Välimäki (2006) underlines that, in fact, the copyright protection is stronger and more competent tool for protecting software than the patent. The granted patent can be seen differently in the national courts and can this way even cause a legal risk. The copyright protection can be more beneficial for daily business operations, because truly functioning markets for patented software are still waiting to happen. Patenting the software can be seen more as a strategic movement. These kinds of strategic questions can be for example: is there a new market on patented industry? Can cross-licensing be beneficial in a new market area? Is there a need to prevent new companies coming into business using patents for eliminating technology standards? Is there a need to weak competitors trying to patent the technology they use? (Välimäki 2006. 89)

4.3 The Other Legal Liabilities

The software product business struggles with many legal liabilities. The copyright and patent acts are the most significant, but trade secret, registered design, trademark and utility models must be also taken in notice. The trade secret was used before the software was taken under the copyright protection. It forbids revealing any business information that can be strategically advantageous to competitors. An employee is obligated be silent concerning the information, which can be classified as the trade secret. The legal responsibility can last even two years after the end of the employment. The trade secret is used usually together with the copyright and patent. It protects the innovation before the actual patent is accepted. On the software product business it is used for protecting the areas, which cannot be covered by the copyright. (Välimäki 2006. 114 - 117)

The trademark is a familiar concept on the software product business. It gives an exclusive right to separate company's own product from other products using a registered format. NBPT summarizes that it "gives you the exclusive right to use the mark as a symbol for goods or services in Finland. An exclusive right means that only the registration holders may use the trademark in their business and may also, when necessary, prohibit others from using their mark or some another mark liable to be confused with it." (NBPT. 2007a) The trademark can for example be a shape, one or more words, letters or numbers. The trademark also protects the services. (Finland's law of trademark 7/1964 1§) The trademark is reasonable for a software product when it is needed to separate from competitors' products or when advertising product's quality or boost up the product's imago. (Välimäki 2006. 117)

The registered designs are used for protecting application user interfaces. In the other words, the registered designs prevent the unauthorized use of the product's appearance. Everyday items are not counted on, but the object must be a concrete article (NBPR. 2007b).

The last noticeable legal registration types are utility models. The utility models are similar like patents, but the common standardization is missing between European countries and in the USA, the utility models are not used. The utility models have lower innovation requirements than patents and "The holder of a utility model has the right to exclude others from commercial utilization of the invention in accordance with his or her utility model. Forms of commercial utilization include such acts as making, selling, using, and importing of a product protected by a utility model … A utility model right may be sold or licensed. In compensation the licensee pays the utility model holder, for instance, a specified percentage of the income produced by the invention (royalty)" (NBPR. 2008). The utility models protect the registered item normally for 10 years. On the software product business, 10 years is more than enough. It is easier and cheaper to get than the patent, but because of the country-specific limitations it has not yet reached wider popularity.

5 The Software License Agreements

Licensing the software with agreements became a standard process during the 1970's, when the huge popularity of microcomputers expanded the potential customer markets. (Välimäki 2006. 139) The copyright protection together with other legal alternatives is an effective way to protect software, but it was not enough. If a manufacturer wants to restrict the use of the software and handing it forward, the separate agreement between the manufacturer and a user is required. (Siivola 2004. 87) This kind of restriction is based on the ideology that the installed application is a copy on the computer's hard drive, nevertheless where the actual physical software use is run. The copyright protection forbids copying the software in any format without permission. Through this logic the software manufacturer can control the use of the software using the separate agreements. (Kulmala 2003. 5)

So forth, it is important to understand the separation of the terms right to use and the copyright. The copyright owning party has a legal right freely to define the possible software development and the user's rights to use protected software. The software agreement has always stronger legal status than the copyright protection. The software license agreement has an individual nature on the legal foundation created by the copyright protection. (Kulmala 2003. 54)

The main perspectives for software licensing can be classified in four categories, which are legal, financial, technical and informational purposes. Usually the mass-market software product license agreements are designed to meet all the descriptions. (Välimäki 2006. 144)

Table 5. The software license agreement perspectives (Välimäki 2006. 144)

Perspective	**Description**
Legal	Agreements define rights and responsibilities between a user and a manufacturer. (The most common situation is that user has only the right to use.)
Financial	Software product pricing model or implementing a new marketing model
Technical	Rules for software development (rights and limitations for further development)
Informational	Political message, legal syntax (expanding programmers reputation)

5.1 The Common Licensing Features

The license is a right to use a single software product or a suite of many software products according to a certain agreement offered by a manufacturer and accepted by a user. The user can be an entity or a single person. The license type is a definition how the product is licensed. It can be counted on two level types, which can be friendly called as a top and a grass level. The top level type license can include many grass level license types.

The top level license types:

1. Company-limited
2. Corporation-limited
3. Network-limited
4. Site-license (unlimited right to use all the software of the certain manufacturer)

The grass level license types:

1. based on the application number (for example per PC or CPU or virtual instances)
2. based on the user number (virtual, simultaneous, floating or individual use)
3. based on the identified users (for example name, unique ID)
4. based on the designated equipment (or the right to use the application only on a certain hardware, Original Equipment Manufacturer, later referred as OEM)

The license types are according to Kulmala's (2003) classification (Kulmala 2003. 58).

The license types describe how the software product is licensed, but what actually is licensed can be generally classified according to table 6. These licenses can include additional services like technical support.

Table 6: The software product version license types

License type	Description
License	Right to use a certain version of the purchased software product (usually includes the right to use earlier versions, but not right to use newer versions)
License + Maintenance	The purchased version includes the right to update the newer version, if appreciable (for example, can be valid only predefined time period or without limitations)
Maintenance	Right to update the software product (for example during the certain time period, can include several updates)
Upgrade	Right to update the purchased version as the current version

The software product license ownership usually is perpetual or it can be subscription or though a service (See the table 7).

Table 7: The Software product license ownership

Ownership	Description
Perpetual	Customer owns the right to use the purchased software on the purchased version level. The license is fixed asset.
Subscription	Customer purchases only the right to use the software.
Service	Customer purchased licenses trough the service or as a service.

Licensing types are usually tight to different licensing models and licensing models can be roughly categorized in the four programs based on the main customer segmentation. The license models are the software products embedded in the hardware (OEM), shrink and wrap products, volume licenses and developer licenses. The software products bought as bundled with the hardware are usually tight to the hardware's life cycle and cannot be reinstalled to another machine. The OEM is usually the most limited way to use a software product and this is normally taken in the consideration in pricing, for example operation systems are bought often as OEM. Shrink and wrap software products are the packet products found on the market shelves. Single downloads from the Internet are also included on the same category. These licenses are not hardware-limited, but the number of simultaneous installations is usually limited. The volume licenses can allow predefined, annually checked or unlimited number of installations. Its ideology is based on the lower unit price when buying licenses in bigger volumes as one-time purchase or during the certain time period. The developer licenses can belong to the previous models, but usually are notified as the own category.

The licensing programs can be seen the customer segmentation-based pricing and conditions in the software use rights. The programs generally are designed for academic, commercial, governmental and consumer use. The same license's unit price can vary according to the used program. Usually the entity has to meet the qualifications set for purchasing under a certain program. The offered software products and the license's use rights under the different programs can also vary.

Table 8. The license model and program summary

License model	Academic (*Schools ect*)	Government (*Municipal ect*)	Commercial (*Companies ect*)	Consumers (*Individual users*)
Hardware embedded (OEM)	X	X	X	X
Shrink and wrap, single download	X	X	X	X
Volume licensing	X	X	X	
Developer	X	X	X	X

5.1.1 The Software for Rent

The basic concept for selling a product is that a customer purchases the product and after this the product is customer's property. This is still a very functioning model, but when buying software, couple things must be considered; the software becomes outdated very soon and new better versions are published all the time. The new software might need big IT investments and because of the fast software life cycle all the companies are not ready for this. The most essential is, why not pay only the software according the time it is used? Solution for this is to use services of the Application Service Provider, so called ASP-services. The ASP as a term has suffered an inflation, but it is still good to be familiar with it. Today's ASP-services are more commonly called as different software or application hosting services. Software hosting means only hosting the use of the software products and application hosting is for hosting the products and the other related services. (Kulmala 2003. 8)

The application service provider offers an option to use the software through the Internet or the private web. The service provider can be a vendor itself or for example, an internet operator, which has got a vendor's permission for this kind of application use and distribution. The customer can be an individual person, company, private or public organization. The customer buys a right to use a specific application from the service provider. The applications and customer created application folders are physically located on the service provider's server and only used though the web. (Kulmala 2003. 8)

The ASP is not for retailed software, but it is a very practical system for the software products. The products which require a lot of processing power, big databases and constant hardware maintaining are ideal for hosting services.

Kulmala (2003) classifies the ASP business related actors in hosting services as Independent Software Vendor ISV, support ISV, Internet Service provider ISP and hosting service provider. ISV is the vendor, who is responsible for manufacturing the software and can also provide some other related services. All these are sold through application service provider. Application Service Provider can have one or more ISV-partners gathered as a segmented customer-fit portfolio. Support ISV is related to ISV, but it consist of different needed components for using the hosted application. Support ISV components can be operation systems or more commonly Virtual Private network (VPN)-tools. Hosting service provider can be ASP's subcontractor as maintaining the required server farm. The bigger volumes the application provider has, the bigger are the environmental requirements so subcontracting this can be a very beneficial. Internet Service Provider is the Internet connection provider. ISP can own the network or rent it from the some other party. Normally ISP is a (tele) operator or with wireless connections a mobile operator. (Kulmala 2003. 10 - 11)

Software as a service (SaaS) is a concept, which is based on the same principles than ASP. When ASP services provide software products applications through the web, SaaS-concept's idea is to service more integrated processes with comparable on-premise applications. An organization uses the software through the web, without owning the actual licenses. The software is compiled in an environment, which is provided by an individual service provider. The billing is measured according to the actual usage or the user volume with different kinds of license subscriptions. SaaS is a functioning model like ASP. Customers do not have to initialize any hardware, licenses and human resources when deploying a new system.

The arising trend after SaaS will be the cloud computing. It will lift the virtualization to the next level. Operation systems, middleware, data stores and application software coupled with grid computing can be bought as service and from the literally from the clouds of the Internet. For now the cloud computing is still vision in Finland, but for sure, it will be the future.

5.2 Against the Mainstream: the Open Source

The free software or more preferred as the open source is an interesting way to develop technology innovations sharing it with others developers or users. One of the most famous results of the open source development is Linux operation system and its version variety. The operation systems are freely downloaded and installed on the workstation without any usage restrictions. The other well-known and used free products are Mozilla's applications and database system MySQL.

The Open Source Initiative, later referred as OSI, founded in 1998, has created the official open source definitions, later referred as ODF, for distributing the open source licenses. ODF states ten requirements, which the application must fulfill. The most fundamental definitions are that the software can be freely distributed and the source code must be included so that it can be shaped or remodeled. The other important feature is that the application cannot restrict any other software and it is a technology-neutral. (Open Source Initiative. 2006)

The definitions are based on the copyright, but oppositely. ODF allow distributing, copying and modifying the software. This kind of approach limits the use of the commercial licensing, but the open source has own license types, which can be also quite binding. (Välimäki 2006. 190 - 191)

Table 9 categories the most popular license types for the open software. *Restricted* is the commercial software and it is included for comparing reasons. It does not have any open source characteristics. The other licenses share partially or all open source features. *Share alike* means that a license do not restrict licensing a whole entity. It actually allows for example linking the components to another entity. *Viral effect* in other hand restricts that the whole entity must license with the same conditions than its individual component. (Välimäki 2006. 192)

Table 9: The most popular Open Source license types (Välimäki 2006. 193)

Criteria	Free distribution	Free use	Visible open source code	Share alike	Viral effect	Network usage
Restricted	-	-	-	-	-	-
Shareware	X	-	-	-	-	-
Freeware	X	X	-	-	-	-
MIT	X	X	X	-	-	-
LGPL	X	X	X	X	-	-
GPL	X	X	X	X	X	-
OSL	X	X	X	X	X	X

Usually *shareware* is a free trial version for testing the software. It does not have any support and some full-product features can be disabled.

Freeware is a term for the software, which is downloaded from the Internet or on the media without any payments or cost expectations. It is, according the term, totally free to use, but the source code is hidden and so it cannot be developed by other parties. The other forms of the freeware are the products, which are free of charge in the private use, but in the business use, the products licenses must be purchased.

MIT is license type created by Massachussetts Institute of Technology (MIT). License allows that the source code is open and visible for the other users, but this is not compulsory. MIT allows distributing the software in a machine code, but even the software is shared as a part of another application under the different license, there must be a mention about the original creators and the permission notice. *BSD (Berkeley Software Distribution)* follows the same principles than MIT, but it has an addition, that the creators' names or the copyright owners are not allowed to use in advertising the application. (Välimäki 2006. 195)

LGPL stands for GNU Lesser General Public License and GPL General Public License developed by GNU Project - free software foundation. GLP is stricter than LGPL and "using the Lesser GPL permits use of the library in proprietary programs; using the ordinary GPL for a library makes it available only for free programs … Proprietary software developers have the advantage of money; free software developers need to make advantages for each other. Using the ordinary GPL for a library gives free software developers an advantage over proprietary developers: a library that they can use, while proprietary developers cannot use it.

Using the ordinary GPL is not advantageous for every library. There are reasons that can make it better to use the Lesser GPL in certain cases. The most common case is when a free library's features are readily available for proprietary software through other alternative libraries. In that case, the library cannot give free software any particular advantage, so it is better to use the Lesser GPL for that library." (GNU Operating System. 2007)

GNU licenses are among the most popular types to license free software. MySQL are for example licensed under GLP.

OSL (Open Software License) is OSI's license fulfilling all the OSI's definitions for the open source.

5.2.1 The Open Source Business Dimensions

The open source can be utilized also in the software business. It offers several different open source licensing models, which can be used as the foundation for a commercial software product. Although all the open source licenses do not allow this. The other possibility is use the open source in customer tailored software project. The costs for the customer and the software business company can be this way remarkably lower. The software business company can also offer services related to the open source software, for example support and maintenance. The IT environments with the open source rarely have the professional resources and if the open source is used with the business critical application, it is extremely important to get the support when needed. The software company can also offer updates for the common errors or value-added services related to the open source.

5.3 The Software Piracy

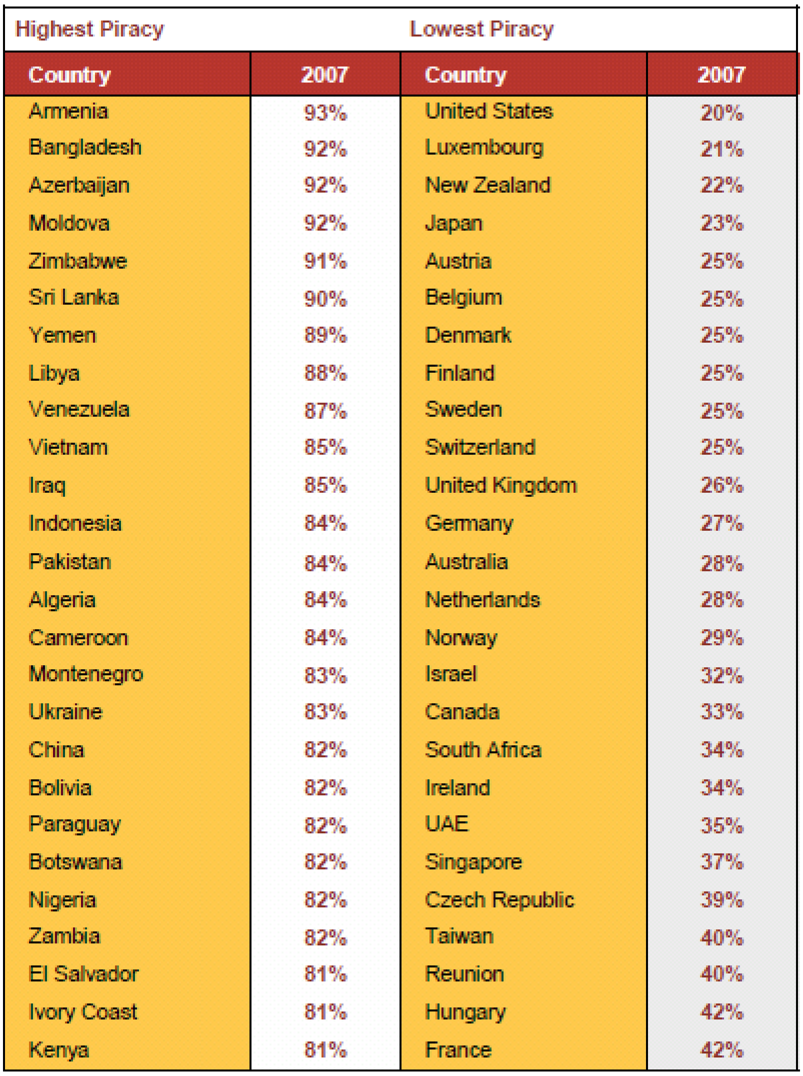
The software piracy as a term, meant a couple years ago only for making a profit with extensive amount of software piracy copies. Nowadays the term covers all downloads and installation without a legal license, even if the purpose is not make a profit. This latest adjust was lobbed by Business Software Alliance, later referred as BSA, and the other same types of associations, supported by manufacturers. As a vendor's point of view the software piracy is most of all, a problem related to software license agreements and through this, related to the software business and the incomes. (Välimäki 2006. 213 - 214)

BSA is an association for the software manufacturers and its partners. It is a nonprofit organization, which goals are to "protect software providers' intellectual property rights, enforce software copyright legislation, and encourage compliance." The biggest members of BSA are for example, Microsoft, Adobe, IBM, Apple, HP, SAP, Symantec, CA and Cisco. (BSA. 2008)

In Finland BSA is a very active institution and it campaigns continuously against the software piracy organizing conferences and bulk e-mails. BSA's most well-known operation is an audit. BSA has a right to demand installation and purchase information behalf of its members about used software in an organization. If the amount of licenses is not corresponding with the software installations, must the audited organization purchase the missing licenses and pay an extra fee for errant software use. This kind on publicity can be very harmful and expensive. In the last hand the managing director is responsible that the organization's software environment is properly licensed. If the organization has global operations, it must be aware other this kind of associations operating in other countries, for example Federation Against Software Theft (FAST) in United Kingdom, also the vendors can and are doing separate audits. (BSA. 2008)

In Finland the software piracy is rare, according BSA the piracy percent is 25 and the founded licensing gaps have been mainly results as the lack of licensing knowledge and careless risk and asset management.

Table 10: PC Software piracy rankings (Fifth annual BSA and IDC Global software piracy study 2007. 4)



Highest Piracy		Lowest Piracy	
Country	2007	Country	2007
Armenia	93%	United States	20%
Bangladesh	92%	Luxembourg	21%
Azerbaijan	92%	New Zealand	22%
Moldova	92%	Japan	23%
Zimbabwe	91%	Austria	25%
Sri Lanka	90%	Belgium	25%
Yemen	89%	Denmark	25%
Libya	88%	Finland	25%
Venezuela	87%	Sweden	25%
Vietnam	85%	Switzerland	25%
Iraq	85%	United Kingdom	26%
Indonesia	84%	Germany	27%
Pakistan	84%	Australia	28%
Algeria	84%	Netherlands	28%
Cameroon	84%	Norway	29%
Montenegro	83%	Israel	32%
Ukraine	83%	Canada	33%
China	82%	South Africa	34%
Bolivia	82%	Ireland	34%
Paraguay	82%	UAE	35%
Botswana	82%	Singapore	37%
Nigeria	82%	Czech Republic	39%
Zambia	82%	Taiwan	40%
El Salvador	81%	Reunion	40%
Ivory Coast	81%	Hungary	42%
Kenya	81%	France	42%

5.3.1 The Software Piracy Types

Business Software Alliance lists five software piracy types, which are the end-user piracy, client-server overuse, Internet piracy, hard-disk loading and the software counterfeiting. (BSA. 2008. What is software piracy?)

The end-user piracy is a piracy type where a user installs intentionally or unintentionally the software, which is improperly licensed or not licensed at all. The most common scenarios are unauthorized media copies, illegal copy distributions, media swapping between organizations and upgrades, which can be used without the actual upgrade license. The end-user piracy can also be purchasing the correct licenses using a lower-price software program, without the required qualifications.

Client-server overuse is installing an application on a network server and too many employees are using it without the prober licensing on the same time or have an access to it. The normal overuse situation could be the application use trough a virtual private network without a prober licensing.

The most common piracy type is the Internet piracy including software downloads from the Internet using unofficial sites, Internet auction sites or Peer-to-Peer networks. Usually this a bigger challenge on the consumer sector than among the companies in Finland.

Hard-disk loading is a risk concerning the purchase channels and their reliability. This hardly is a problem in Finland, but in developing countries, where the laws and regulations are not yet taken place in a society, it is more common phenomenon.

Software Counterfeiting can be seen more as a traditional piracy way of selling and buying software forgeries. This piracy type includes all the counterfeits on the packages, manuals, certification stickers or stamps or media. Normally this kind of piracy is easy to recognize, because the suspiciously low prices.

Verifying that the used software really is legal and bough through the rightful channels can sometimes be a real challenge. The best practice is to always save the purchase invoice and the license certification. License certifications can be very different depending on the manufacturer's policy. Some vendors send the certification by email and some by paper. If a license is purchased though a license volume program, the licenses normally are added on the vendor's agreement management site. The purchase receipts can be received also from the reliable reseller.

6 The License Management Challenges

The variety of different software applications can be remarkable today's IT environments and direct and indirect licensing costs can be even a half from an organization's budget. The applications need constant monitoring and maintaining, not forgetting the detailed and on-time license management. Unfortunately the license tracking can be challenging and ensuring gathered information's validity and integrity, it can be even impossible. The common scenario is that organizations embed the license management to the overall IT asset management. This can be a functioning model, but in most of the cases the nature and the business practices of licensing are not supported well enough. The license life cycle, discussed more later on, is unique for previously introduced legal liabilities and licensing features concerning to intellectual property rights.

6.1 What? Where? How many?

The common challenges of license management can be summarized in the following questions; what software licenses our organization have? Where the corresponding installations are and what other applications are running in our organization's environment? Do I have enough or too many licenses? Are all the licenses in use? The more complex IT environment is, the more challenging is to find reliable answers.

Without appreciable tools the purchased license information can be too human-depended. The license management database including also the exception knowledge can just quit. The common situation is that the software licenses have been bought from different sources using several license types. Without a reliable data source all this information is challenging to maintain reasonable and it can be that the licenses must be traced from the beginning. The operation can be afterwards time-consuming and generates extra costs in form of used resources and working hours. The gathered information does not ensure that the current installations are corresponding. The vendors also assume that the possible physical proofs like certification papers, stickers and stamps or media are properly stored and found if the organization is audited.

For validating that the current installations are compliant with license assets, perhaps the separately inventing tool for the software inventory is required. The tool should trusted cover at least all the installed commercial software programs. The manual separation between licenses and the installed software versions can be timing, but is a required operation for ensuring the real compliance. The challenges with this process can be the lack of time, resources or basic licensing understanding.

Many organizations have limited basic users' rights to do installation. This prevents effectively unwanted entertainment, hostile program, and piracy installations and, most of all, installations without a license, but it does ensure that the installed business programs are really in use. The application might be installed from historic reasons or purely the software is used only by a marginal employee group. The organization might be purchasing licenses that could be utilized from the own environment.

All these challenges are emphasized in organization fusions and separations. If the existing license management has been poorly maintained, the costs can increase significantly. The most of the manufacturers require the official notice of the perpetual license transfer. The license transfers can require detailed data for a longer time period.

6.2 Licensing Related Risks

It can be said that the license management is risk management. The biggest risks related to licenses are that the organization has not purchased enough licenses and it is punished for that or the organization has purchased too many licenses and the investments have been in vain. The more potential scenario is that if the organization does not have any monitoring system, it probably purchases chosen the second option, just in case. The risks grow remarkably if the IT environment and the related processes are not under organization's control including users' installation right limitations. (Jordan & Silcock 2005. 245)

Referred as International Organization for Standardization software asset management standard ISO/IEC 19770, the negative license gap can arise a risk of damaging the public imago, which can cause in long-term more damage than a penalty. The companies operating in certain reputation-based business sectors, for example banks and insurance companies and strategic consulting companies can be very vulnerable to this kind of risk. The poor license management can increase the risks of IT service interruptions and IT service quality deterioration according to ISO/IEC 19770. ITIL as other common standardization system includes license-related risk software over-deployment.

The most used standards related to the license management are ISO/IEC 199770 Software asset management, ITIL along with lighter COBIT. The standards are discussed more accurately on the software asset management - chapter.

7 The Software Asset Management

The minimum definition, when discussing only on the license management, is that the organization has black on white what applications are running on organization's hardware and why. After this the effective monitoring with required operations can be built (Jordan & Silcock 2005. 244). The required operations are more concentrated on the corporate governance, which is for what the software asset management is developed. The software asset management, later referred as SAM, is managing the licenses with license compliance (license management) and processes, operations, business practices and policies related to software assets utilized through the whole software asset life cycle.

The life cycle of the software asset can be summarized to the different phases, which are procure, distribute, maintain, monitor and dispose the software asset (see the Figure 13). The golden rule plan, do, check and act should be utilized in every phase insuring the functionalities of the life cycle management.

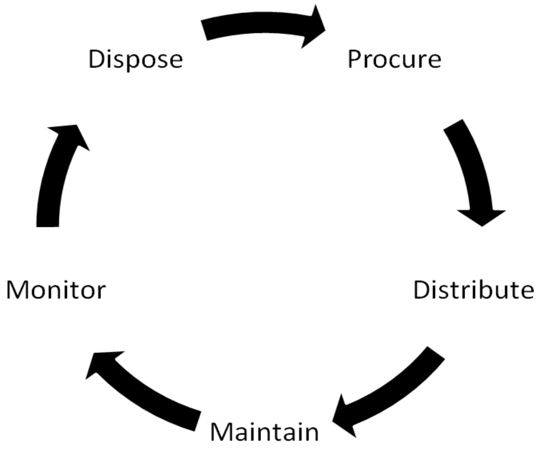


Figure 13. The software life cycle (BSA. Guide to Software Management)

7.1 The Scope of SAM

The software asset management is managing of the organization's software assets, which can be classified as *utilized software, purchased software licenses* and *physical software license assets.* In addition SAM is also for managing *Users* related to the software use.

Utilized software is the software which an organization uses consciously or not. The software is installed on the organization's hardware (computer, laptop, server or mobile phone) or it is accessed through private networks or Internet. Utilized software includes all the installed software nevertheless it is executed or not.

Purchased software licenses are all the licenses purchased by the organization during its life cycle. The proofs of purchased software must exist in both forms; vendor's electronic or physical certificate or the purchase must be found from vendor's database, the required transaction from customer must be proven with an invoice or assimilate record. Purchased software licenses are expected to be utilized according to software product use rights.

Physical software license assets are the physical proofs of the purchased software. Physical assets are expected to distribute according to the conditions of the software product use right. Normally physical assets are certifications papers, stamps, stickers, boxes, installation and toolkit CDs and DVDs.

Users are all the organization's employees or personnel, who have an access to the organization's software assets. Software Asset Management includes the procedures, processes and policies, which the users are expected to follow. Procedures and processes are action models for the phases of the software life cycle. Policies are guideline for the operations. The action models and operations should be defined according to the organization's operational levels, for example management, finance, IT services and other departments.

7.2 The Objectives of SAM

The objectives of SAM can be divided on external and internal objectives. The external objectives describe the software asset goals for whole organization. Those can be seen as achieving a better risk management, cost-effectiveness and competition readiness based on the on-time knowledge. The internal objectives concentrate on internal operations on departmental and location-based level. If the internal objectives are achieved as set or set overall correctly according to organization's business operations, the external objectives should be achieved automatically.

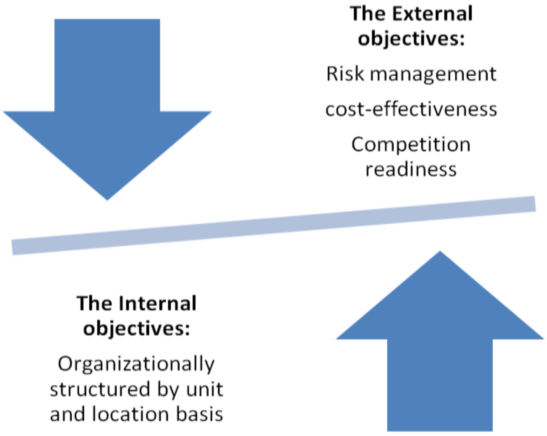


Figure 14: The objectives of SAM

When investigating the external objectives more closely the risk management can be seen consisting of the risks generated purely by external pressures. These kinds of risks for example could be a license audit performed by Business Software Alliance based on a random selection or virus managed to bypass firewalls. According to the software asset management objectives, if achieved, the audit's result is positive and the publicity following can be a benefit to the organization and the virus is identified and removed before a larger damage.

Cost-effectiveness is achieved when software asset management related processes are functioning without any delays. Software and a license are acquired as defined and the user is capable to use the demanded software when needed. The basis is that the license programs for the lowest license unit costs are decided on the management level before according to the organization's software behavior and the future's road-map. The behavior includes the software usage. All the purchased software is expected to be in use and if un-used installation is noticed, the license is drawn back for re-allocating, instead of a new purchase.

Competition readiness is for improving defining, planning and implementing software life cycle related issues like roll-ups, application retirements, changing the system platforms or the used applications. The software asset concerned decisions made on the management level should be based on the reliable information. The information should be updated frequently for more accurate forecasts and budgeting.

The Internal objectives are based on the organizational structure and the organization's location or locations following the departmental share of responsibility. In practically this means that software asset management as a concept is understood through the whole organization and the ownership of the SAM processes is appointed. The SAM processes are defined more specified on the next chapters.

The internal objectives can be separated between an operational, financial and managerial sector. The operational level's SAM responsibility could be performing the actual software inventory ensuring the data reliability and consistency. Through this the operational level is also responsible for maintaining the software environment required for an organization's daily operations securing the internal customer satisfaction. The internal customer satisfaction is critical and directly measured by the work efficiency. The systems and the applications should be defined and identified on the management level as well as the purchase programs and purchase channels on a financial level. This way carefully planned SAM processes reduces the costs and grow the level of the producing work.

When the set internal objectives are failed to meet, the external objectives can be jeopardized. The extreme valuable information for the organization can be revealed, which can cost unmanageable pressures from the external parties.

7.3 The Structure of SAM

When the organization is structuring the software asset management it should investigate and measure the following factors: purchase, software policy, software ethics, inventory, audit, business requirements and global processes. These areas are tightly connected to the successful SAM and are used when building SAM in use. The SAM structure is used for setting a right level to the organization's software asset managing. All the factors are not effecting on the same level to every organization and some of the factors are not even needed to be investigated. When structuring SAM the ultimate value is to meet the objectives.

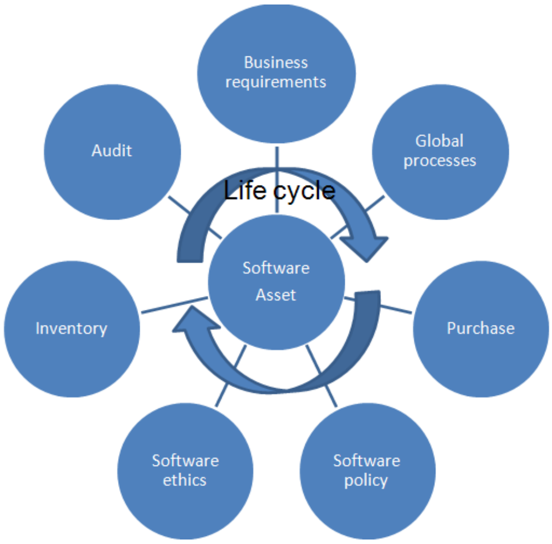


Figure 15: The structure of the software asset management

The software asset in figure 15 reflects the organization's existing software assets. These assets can be categorized as licensed, free and forbidden software really running in the organization's environment, purchased licenses information separately or embedded in the hardware as on paper or in any financial knowledge base. The licensed software is defined here as the installed software, which according to the vendor's requirements demands a license. Nevertheless what the type of the license is. The shareware in counted in this category and the software freely installed business commercial programs, like WinZip, which is free only in private use on a private computer. Free software is freely installed software, for example Adobe Reader. It does not require any license, but it is needed for organization's operations.

The forbidden software is defined according the software piracy types, which where end-user piracy, client-server overuse, Internet piracy, hard-disk loading and the software counterfeiting. This type of installations can also be harmless for organization's imago, but decrease the work motivation and eat the hardware resources like games or other entertainment applications. The forbidden software running in the organization's environment is always a risk in some level.

Purchased license information are all the licenses proven to be purchased according the organization's own accounting with existing invoices and manufacturer's certifications. The use right can also been bought as OEM, when the proof along the previously mentioned is a proof of the purchased license.

The software asset should be managed according to the software life cycle. In this structure the software assets are not including disposed or future's acquisitions.

In the SAM structure the software purchases should always be done with clear understanding who is allowed to purchase the software, why the software is needed and how the purchase is done. Who means that the organization does not have the overlapping purchase processes and the purchases are always centralized, if possible. In the worst scenario the organization purchases the same license accidentally twice or licenses are acquired through different programs, which complicate license maintenance and the license unit price can be higher than it would be through the centralized program. Why describes the knowledge of buying a right license type for an organization's configuration, for example, is the license needed per processor or per user. In the last hand a customer is always responsible that the purchased licenses are correctly bought. Purchases should be contrasted with installations so that the license amount is same than the number of installations. The proof of purchased license should always be stored in centralized electronic and physical storage, where those can be easily fetched. How the purchase is done specifies the used license programs for every vendor and the used reseller or resellers.

In an ideal situation is when the management level has identified the software critical for daily operations and understood why the certain applications are chosen. What is the business value for used technologies and applications? This restricts the application variety with similar functions easing IT support, improving IT service quality and possible enabling the use of more beneficial volume program. This is the SAM structure's software policy. It goals are to share knowledge of the used application between the users in the operational and management levels. If the user understands what are the allowed and standardized software applications and why, the risks of unintentional misuses can be minimized. The management, behalf, understands what kind of different software use is demanded for the organization's operations. Software policy also includes beforehand chosen cost-effective purchase programs and co-operative channels simplifying and fastening the purchase processes from the operational level to the management level and from the management level to the operational level. The software policy also defines how the physical and electronic license asset is stored.

Software ethic in the SAM describes the user's common software behavior and what is the role of the organization to educate and teach the software use and policy understanding. The user is responsible for following the organization's software policy. The user should know what kinds of agents can be a secure risk and what type of applications are not allowed to run company's workstation. The users should, however, have all the required work-oriented applications running smoothly on their computers. If the user needs IT support or a new application, the user should know where to contact and how long it takes to an action.

Inventory is a database for the SAM, where installed and used software application versions and purchased licenses are stored. It maintains the license compliance tracking that there are enough licenses compared to the amount of installations and it ensures that free licenses are allocated to new installations, without any new purchase. When the database is according to IT/SAM, it also has information about the application usage. This ensures that the installed applications are really used and the license is not wasted, when the hardware is disposed. The installation data is gathered using the tools appropriate to the SAM. The main issue is that the tool should be trusted and collect enough data, other way the database is not reliable enough and its integrity can be endangered. Different tool types are discussed later on SAM processes. The database should be kept up to date so that the software scanning should be performed repeatedly in a predefined frequency. The database is the core component, when the financial, management and operational reports are produced. These reports are essential when planning the future investments and budgeting and ensuring the objective fulfillments. The accurate reports also can give a negotiating advantage with vendors and resellers.

An audit in the SAM can be seen in two ways, it is the audit performed by an external party or it is auditing internal processes and policies for better performance and risk management. The manufacturers and other alliances, like BSA can perform the external audits based on the copyright, other the related laws and the license agreements. Audits can also be performed by different international standardization organizations when the organization is obligated to follow the standards. The most used standards among the organizations are Information Technology Infrastructure Library (ITIL) and International Organization for Standardization (ISO). ITIL has developed best practices in IT management and framework for IT Service Management. British Standards Institution's Standard for IT Service Management (BS15000) and ISO 20000 support and is supported by ITIL. (ITIL. 2008) ITIL has developed also own processes for software asset management.

ISO is the largest developer and publisher for international standards and has developed ISO/IEC 19770-1:2006 for software asset management. It describes and defines standardized processes for the software asset management to satisfy. ISO/IEC 19770-1 concentrates more on what information is needed for satisfying the corporate governance and support for IT service management. It does not define so far how the software asset management should be conducted. ISO has published only the first part of this standardization. The part two and three are not yet finished. (ISO. 2008)

The audit contains also country specific regulations, which regulate the software assets. These regulations are followed by the whole organization, despite the operation locations or only the certain locations. This kind of governmental requirement is for example, the famous, Sarbanes-Oxley Act of 2002 (SOX) in the USA. It regulates the United States financial reporting and accounting systems. According to U.S Securities and Exchange Commission "The Sarbanes-Oxley Act of 2002 ("the Act") sought, among other things, to improve our system of financial reporting by reinforcing the checks and balances that are critical to investor confidence." (Study Pursuant to Section 108(d) of the Sarbanes-Oxley Act of 2002 on the Adoption by the United States Financial Reporting System of a Principles-Based Accounting System. 2003) It does not yet have a strong impact in Finland, except some global companies, which operates in the USA. The European Union 8th company law directive, called also "EuroSOX" will be probated beginning of the year 2009. It has SOX type regulations covering also the software asset. (DIRECTIVE 2006/43/EC)

When discussing audits as the internal processes, those help to learn and understand the software environment. The internal audit can help the organization to achieve better operational and informational processes. This can save time and costs removing irrelevant and time-consuming software asset related operations.

The business requirements are more the high-level SAM. These requirements answer for effective resourcing, professional employees and standardized processes in the different business sectors. The defined business requirements should prevent conflicts between the user and business. The users might be accustomed to use a system, which is not integrating enough with other business applications or the systems are so old and limited that users have challenges to perform their tasks. This slows down the productivity and can cause extra costs to the organization. Employees should also be trained for used applications for ensuring the full effectiveness offered by the used software. The organizations can have a very high mobilization, traditional office environment, product development, production development or a mixture of those. The licensing needs can be in every environment different. The business requirements include the organization's IT road map for forecasting the best license deals in the future.

Global processes are also the high-level SAM. The global processes are for creating a common easily manageable concept for geographically and cultural different offices or domestic, if it is the operational area. Defining the global processes can help obtain the vendor's international agreements and benefiting through it, for example some countries cheaper currency value or different product use rights. Global processes are also related to the governance compliance and its different country specific regulations.

7.4 SAM as a Concept

Software Asset Management (SAM) is managing the organization's software assets from all the perspectives for achieving the set objectives. When it is discussed as a concept, it should fulfill two conditions. First of all, it should have an existing, documented strategy with named processes and process ownerships with a clear scope and objective statements. Secondly, it should be continuous and continuously improved and controlled, closely tight to IT asset management, but as an individual system.

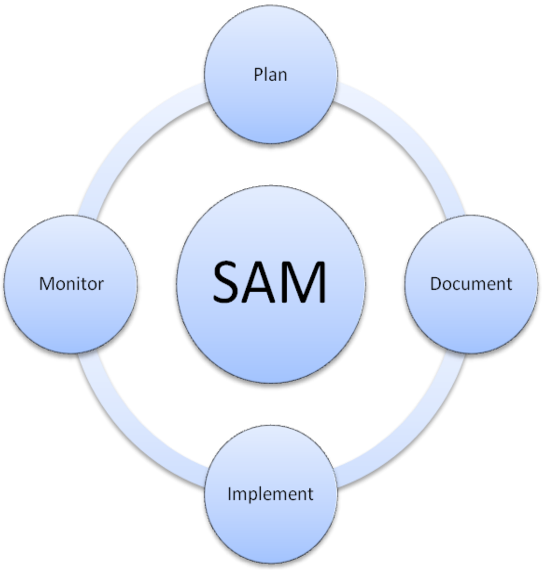


Figure 16: Life cycle of SAM

The life cycle of SAM is consists of planning, documenting, implementing and monitoring the SAM. Planning the SAM for the organization is the most essential stage and it should be done with awareness. The planning stage includes creating the external and internal objectives, defining the SAM scope on item level and the possible SAM tool used. The plan defines also the outcomes of the SAM, for example what data is needed and how often for the objectives. The process ownerships and responsibilities are planned before the documentation for every process level.

The Documentation is the proof of the SAM and it is necessary for continuous processes. It defines the final roles and processes for the SAM. Without a proper and properly updated documentation SAM could be too overwhelming to maintain. Documentation minimizes the risk of the SAM interruption and information backups. The interruption can occur when new software or hardware is distributed or when old software and hardware is disposed. When SAM is bought as a service from the third party and a service provider is changed the documentation has a very important role for guiding the new service provider. The documentation should describe the roles and responsibilities so well that when a new entitlement is appointed responsible as the SAM processes, there are no delay or interruptions in SA processes.

After the SAM is planned and documented, it is implemented according to the documentation. The operational processes are appointed to the named entitlements and a tool for SAM is installed, if needed. The tool helps to gather needed software data for the SAM. In practically the implementation means that the data of installed software and purchased licenses is combined and that data is utilized according the SAM purposes. Employees are instructed for SAM and they understand its role and their role in it. The implementation includes the separate financial, service level and security management, if needed.

After implementation, SAM tool should be monitored for errors and malfunctions and the processes should be validated. Monitoring can be seen as the operating SAM ensures its functionality. SAM should be improvement when needed and it should not be time-consuming or heavy process to keep running. The monitoring includes the processes for the software life cycle. It should support software procurement, distribution, maintenance, monitoring and dispose.

7.5 The Processes

The accurate processes are the key elements for functioning software asset management. The process levels are caricaturized as management, purchases and operational. The ownerships of the management level processes belong to the organization's decision responsible entitlements, for example different levels in finance, legal, lower level management and human resources. The purchase - level's process ownerships belong to entitlements responsible for license purchases and the operational level processes belong to IT department maintaining IT services and all the individual employees.

Every process level is responsible for keeping the SAM life cycle functionally. Planning and documenting before the actual SAM implementation is done on the management level. The actual implementation phase and updating the changes on the lower level processes are executed on purchase and operational level. The SAM monitoring is done by all the levels. The management level monitors are the set objectives met, purchase level monitors the license related changes and the operational level monitors the system.

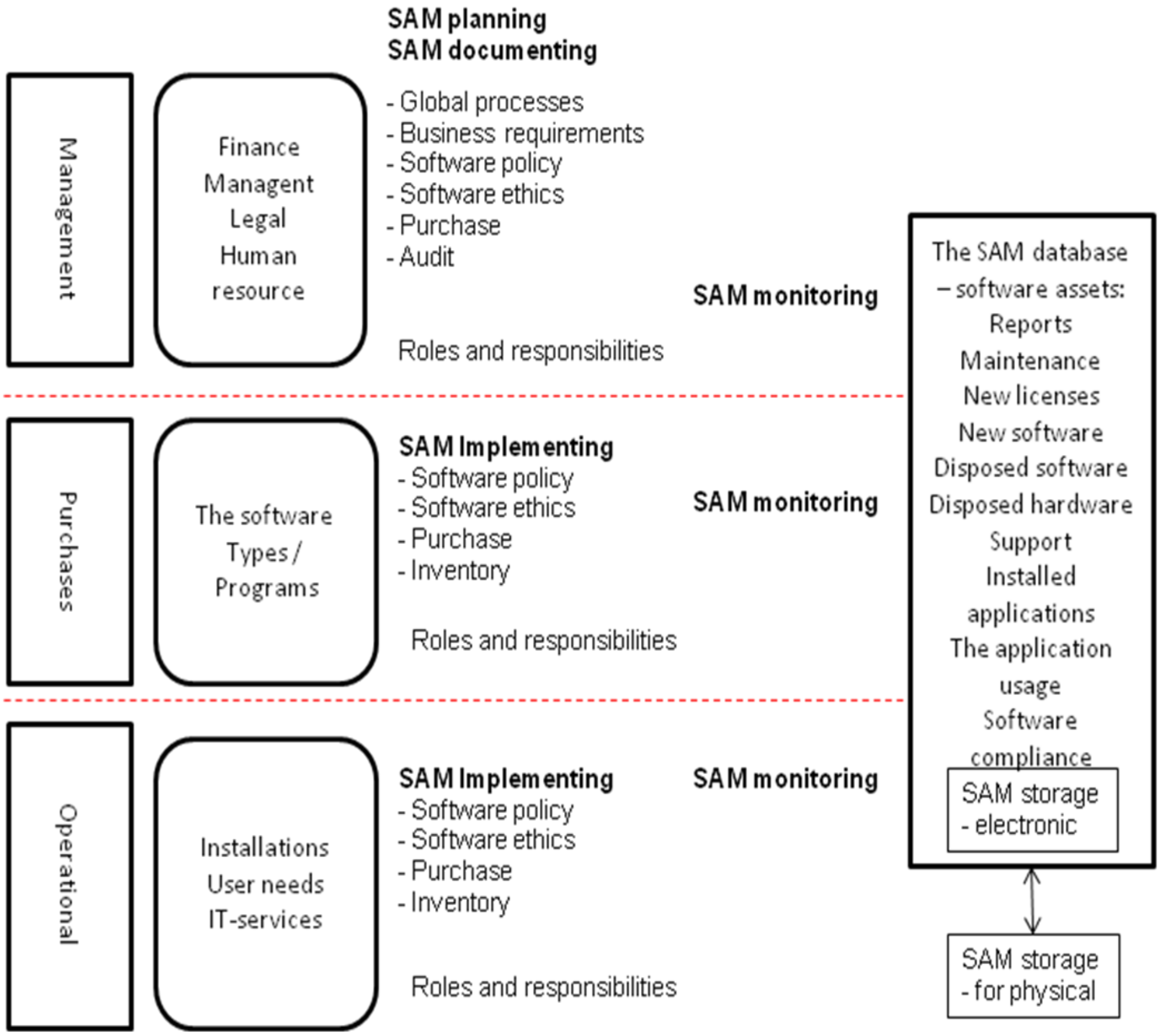


Figure 17: The SAM process map

The software database has the inventory results of installed applications. It has up-to-date or even real time knowledge of the software environment. It should be updated automatically or manually when a new software or new hardware with installed software is acquired. If the software is disposed, it should be removed from the database. If the hardware with installed software is disposed, the licenses related to it, should be free for new installations, if appropriate. The purchase level processes are responsible, that new licenses are input to the database and allocated and the certifications are stored properly. The operational processes are responsible that the physical software assets are stored properly.

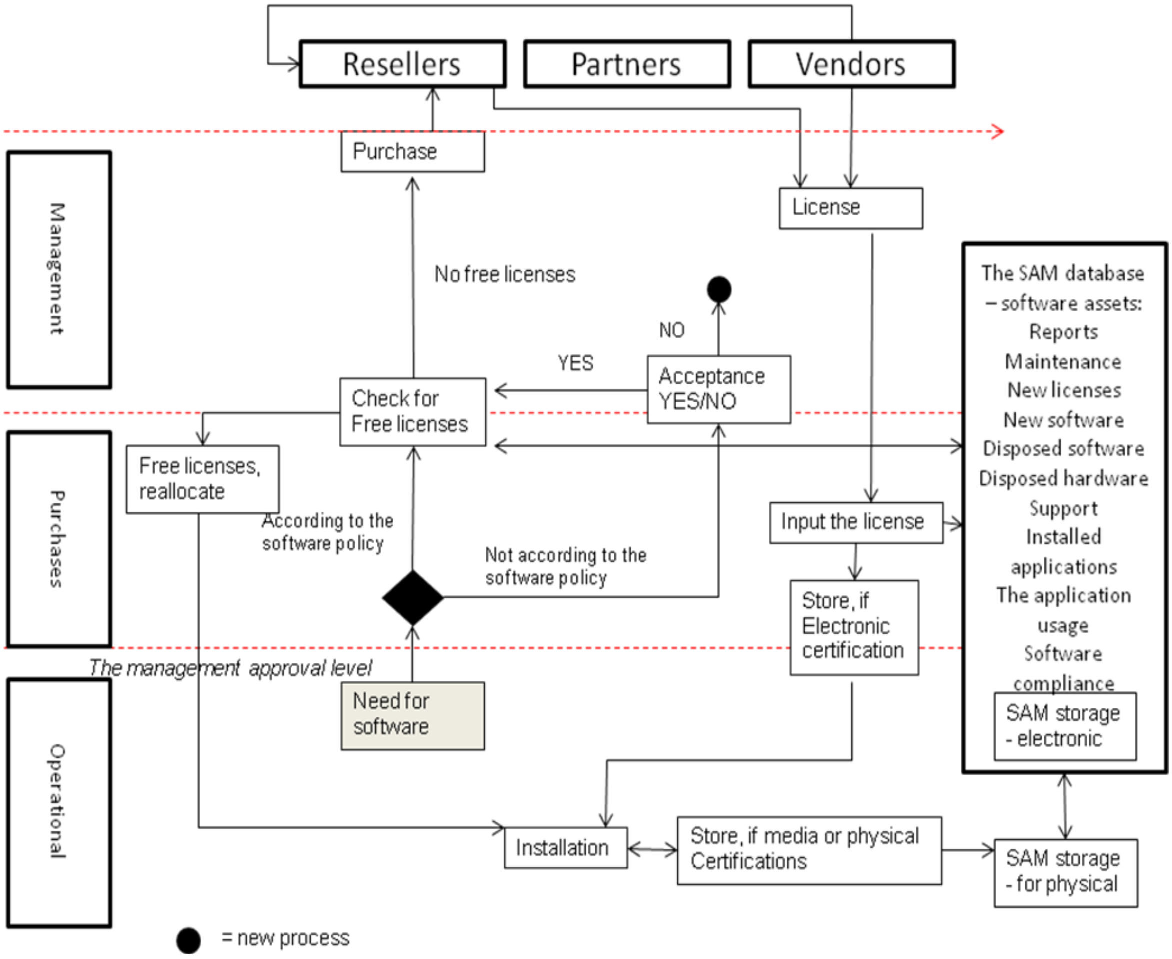

Resellers
Partners
Vendors
Purchase
License
Management
No free licenses
NO
YES
Acceptance YES/NO
Check for Free licenses
The SAM database – software assets: Reports Maintenance New licenses New software Disposed software Disposed hardware Support Installed applications The application usage Software compliance
Purchases
Free licenses, reallocate
According to the software policy
Not according to the software policy
Input the license
Store, if Electronic certification
The management approval level
Need for software
SAM storage - electronic
Operational
Installation
Store, if media or physical Certifications
SAM storage - for physical
= new process

Figure 18: The example sub process for a new license purchase

The sub processes are for supporting a software life cycle and can include several other parallel processes. Figure 18 demonstrates an example sub process for procurement of a new license.

7.6 Tools

The software asset management can be very challenging and time-consuming with many manual work processes without a prober tools. When defining a tool it is essential to understand on what level the software asset management is needed to be in the organization. It is no need to purchase heavy systems if the organization's software environment and license needs are light. The possible scenario is using several tools with different main functionalities, but this it should be ensured that this does not break the SAM integrity or validity and also that the processes are kept simple. The other scenario is use only one tool with required functionalities for the SAM.

The main functionalities for tools are discovery, license asset management and metering (see the table 11). The tools should help getting needed reports and offer a possibility to design own reports. If the tool automatically includes the data, which is needed to be updated for the SAM database, the updates processes should be validated. The practical tool can be integrated to the organization's other systems for getting more accurate date, for example according the cost centers, department and locations. The tool can also be proactive, for example sending automatically reports or alerts, when the action is needed.

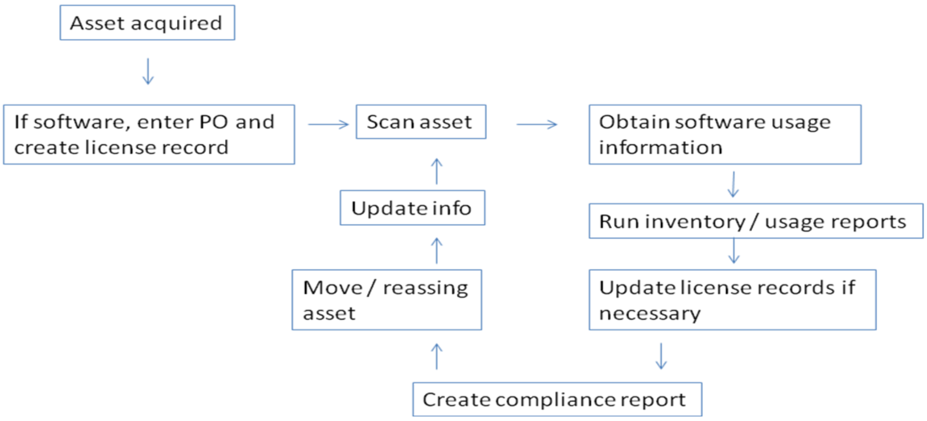

Asset acquired
If software, enter PO and create license record
Scan asset
Obtain software usage information
Update info
Run inventory / usage reports
Move / reassing asset
Update license records if necessary
Create compliance report

Figure 19: Novell's tool ZENworks asset management example related SAM processes (Novell, Novell ZENworks Asset Management, Advanced technical Training, 2007)

Table 11: The SAM tool functionalities

Tool	**Purpose for SAM**	**Functionality**
Discovery (security)	Software inventory of installed software. What is installed and where it is running and who is using it?	Recognizes the number of installed software versions with manufacturers, software users (user, hardware) on different platforms (ex. Microsoft, Mac). Recognizes possible hot fixes and security and normal updates. Recognizes possible the software used on virtually or over the network. Scan possible the network. Possibility to scan offline hardware with separate program.
License Asset Management / Inventory (procurement)	Software agreement and licenses. What licenses owned by whom and where? Are there too much licenses or too few? License compliance.	Software agreements and licenses. Possible license allocation to the asset (hardware, user, cost center, department…) Electronic proofs. Recognize possible the used license keys (tracks licenses). Recognize possible already the software needing a license (suites and single). Recognizes possible allowed version variety with a single license. Frees the licenses from the disposed hardware. Possible license costs for reports.
Metering	The software usage. Is the (licensed) software actually used?	Recognizes the installed software usage (user, application, average). Recognized possible the virtual application usage or over the network.

8 SAM as a Service

The software asset management is a commonly outsourced service today. Nevertheless it is often involved in the organizations core processes. Outsourcing can also be a good opportunity to leave the SAM life cycle's processes including possible technical support to a trusted third party's care. Often the organization does not have the resources or time, the licensing and SAM knowledge needed for functioning SAM. SAM can be a burden and its costs can be bigger than its value if it is not planned well and correctly. The challenge to SAM as a service is to ensure that a service provider has enough information about organization's operation environment and understand its special features (.Sipilä 1996. 27). For this reason SAM can be also requiring service for service provider's to provide. The decision for a customer is that is the whole SAM outsourced or just part of the processes and for the service provider to decide if the whole service is wanted to offer or only supporting pieces, for example an asset inventory.

8.1 The Service Provider's Point of View

The SAM as a service is the challenging service to provide. It requires that the service provider has own resources or partners for the required licensing specialism, understanding of the SAM processes and a ability to understand the SAM related core factors in a customer's operation environment. The customer organization must have a maxim value and the service provider needs to meet set target profit from the service (Sipilä 1996. 12). The SAM service provider should carefully plan on which level of productization is going to operate on and based on that, what SAM operations it wants to produce or does it modularize different services or does it produce the SAM as a whole concept. The productization level effects on the needed resources and processes to be allocated on the service and through this has impact on the service's pricing.

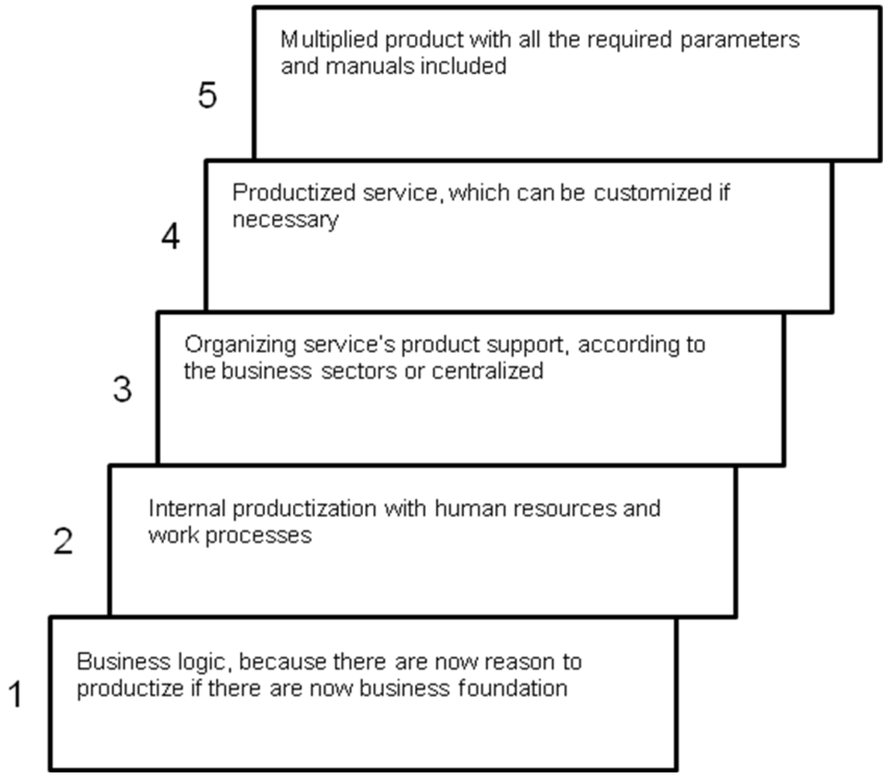

5
Multiplied product with all the required parameters and manuals included
4
Productized service, which can be customized if necessary
3
Organizing service's product support, according to the business sectors or centralized
2
Internal productization with human resources and work processes
1
Business logic, because there are now reason to productize if there are now business foundation

Figure 20: The productization levels (Lipiäinen 2000. 301)

The SAM as a concept is strongly related to organization's different areas, which open the possibilities to offer several support services or separately charged additional services. The support services can be wider licensing consulting improving the service overall quality. The SAM processes do not include the vendor-related agreement logic or understanding of different software programs and the program features. This kind of up-to-date knowledge adds remarkably the overall value for SAM and can help customer to save more licensing costs. The additional services can be for example helpdesk, software solution projects, roll-outs and different IT environment and service surveys.

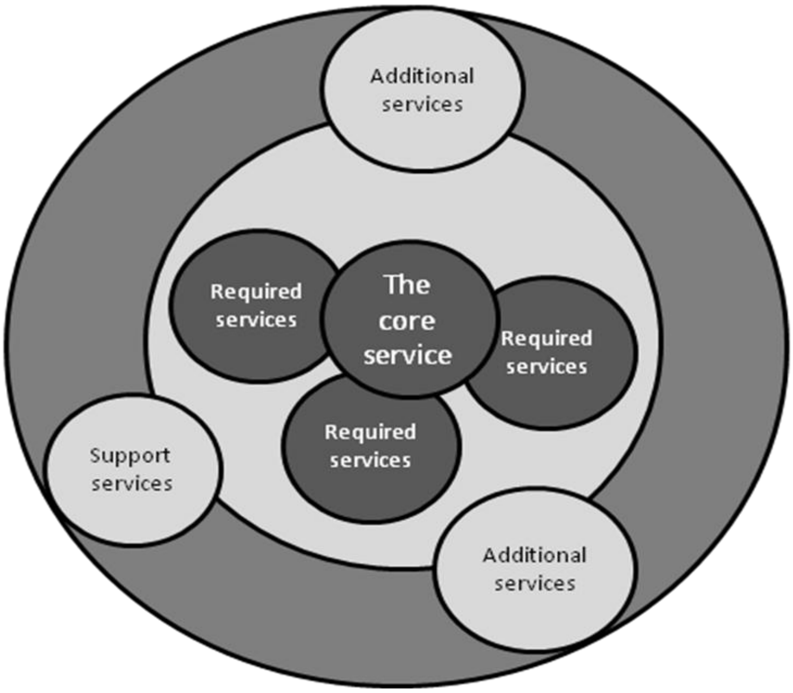


Figure 21: The core service and different support services (Sipilä 1996. 64)

9 Conclusions

The first challenge has been and will also be in the future outdated study books and the availability per-reviewed resources concerning software asset management, because it usually is the confidential property of the enterprises and is based on the standardizations, which have a very strictly limited use and contents. The articles level information is widely available, but behind the articles are usually the commercial purposes.

The other challenge was unspecified boundaries between software business, software product business and the related services. The categories besides Eero Hyvönen's (2003) edited book used in this research were defined based on the colleagues opinions and own practical needs

This research is written to answer and give sufficient set of tools to understand to the software product environment and the actors working in it in Finland today. The shortened life cycle of the software products and break-through innovations has reshaped the markets to concentrate more on the tailored service solutions and volume products. The competition in the software product business and the low margins limit the profits and forces the actors reorganize and form the better services to be more fitted in to continuously changing situations. At the same there is a huge need to develop own earning logic according to the product lines with the better profits and sell more globally common solutions. The current economical crisis will speed up the changes and there is a big opportunity to eat away the smaller actors in the branch.

The common principles and legal aspects of licensing have a very strict foundation on the copyright and the agreements between the manufacturers and the end-users, but this traditional view is insufficient, because of the new over-the-web delivery, like software as a service (SaaS) and cloud computing. The customers are unwilling to pay for these new ways using traditional software product licensing pricing and these new approaches forces manufacturers, distributors and resellers to develop new ways to their licensing policy and earning logics. At the moment the software asset management is, both, the challenge and the answer for these new situations.

The software asset management is a complicate concept and in it, is related so many aspects. The organizations need specified persons with training and education to understand and manage all these details. The software asset management (SAM) needs very comprehensive and total understanding of the processes inside of the organization or as an outsourced service from a reliable reseller or specialized support service provider.

The global view of the software product business and the open source are so wide that there is a recommendation to do separate study in on those in the future, also the software asset management (SAM) as service needs deeper impact by someone who needs more specified information on that business area.

10 Discussions

Key learning of the software licensing environment when reflecting to the software asset management is that the difference of the total costs and savings for the small and middle-size companies are small in economical value, but the importance of the purchasing and licensing policy will increase significantly when the size of the organization increases. For example the organization with size of 8000 workstations can save easily 10 - 15 % of the spent amount of money on direct and indirect licensing costs. The saving means 1,5 millions spending company 100 - 150 000 Euros yearly.

In the near future the companies will increase to pay attention to all additional costs due to the economical crisis and expanding costs of labor and software. The solution for organizations could be hiring dedicated persons for managing the whole software asset management concept. At the moment a typical way is to handle the licensing management processes besides on the main responsibilities and without a proper training or education or even knowledge or understanding of the licensing principles and software behavior in their own organization.

As another solution could be the continuous research to get best benefits of the new purchasing models available together with the reliable software providing companies. The traditional licensing agreements do not necessary support rationally the organizations' IT-environments and this can cause unexpected hidden costs and waist of resources.

REFERENCES

Adobe 2008. Adobe's licensing programs and discount levels. For Reading: http://www.adobe.com/aboutadobe/openoptions/overview.html Read: 26.9.2008.

Business Software Alliance. 2008. About BSA & Members. For Reading: http://www.bsa.org/country/BSA%20and%20Members.aspx Read: 19.10.2008.

Business Software Alliance. 2007. Fifth Annual BSA and IDC Global Software. For Reading: http://www.bsa.org/country/Research%20and%20Statistics/Research%20Papers.aspx Read: 19.10.2008.

Business Software Alliance. 2008. What is Software Piracy. For Reading: http://www.bsa.org/country/Anti-Piracy/What-is-Software-Piracy.aspx Read: 20.10.2008.

Business Software Alliance. 2008. Guide to software asset management. For Reading: http://w3.bsa.org/finland/press/ Read: 14.10.2008

The European Union. 8th Company Law DIRECTIVE 2006/43/EC

Finnish Law. Copyright Act. 8.7.1961/404 1§ 5§ 6§. 24.3.1995/446. 11.1.1991/34.

Finnish Law. Trademark Act 7/1964 1§.

The Finnish Funding Agency for Technology and Innovation. 2008. Tämä on Tekes. For Reading: http://www.tekes.fi/tekes/ Read: 23.9.2008.

The Finnish Funding Agency for Technology and Innovation. 25.5.2006. Tulokset. For Reading: http://akseli.tekes.fi/opencms/opencms/OhjelmaPortaali/ohjelmat/SPIN/fi/tulokset.html Read: 23.9.2008.

The Finnish Funding Agency for Technology and Innovation. 2008. Presentation material. R&D in Finland. For Reading: http://www.tekes.fi/eng/publications/presentation_material.html Read: 24.9.2008.

The Finnish Funding Agency for Technology and Innovation. 2003. SPIN - Software products 2000 - 2003. For Reading:
(http://akseli.tekes.fi/opencms/opencms/OhjelmaPortaali/ohjelmat/SPIN/en/etusivu.html)
Read: 24.9.2008

Free Software Foundation, Inc. 2007. Why you shouldn't use the Lesser GPL for your next library. For Reading: http://www.gnu.org/licenses/why-not-lgpl.html Read: 17.10.2008.

Haikala, I. Märijärvi, J. 2002. Ohjelmistotuotanto. 9th edition. Renewed 2002. Talentum. Helsinki. 2003.

Hansel Oy. Welcome to Hansel's Internet pages. For Reading:
http://www.hansel.fi/index.php?id=286&action=empty&lang=en&s=1
Read: 27.10.2008

Hietanen, O. Nurmi, T. 2005. Ohjelmistotuoteliiketoiminnan osaamiskeskus.

Hyvönen, E. (toim). 2003. Ohjelmistoliiketoiminta. WSOY. Porvoo.

International Organization for Standardization. 2008. ISO/IEC 19770. For Reading: http://www.iso.org/iso/ Read: 20.10.2008

Information Technology Infrastructure Library. 2007. For Reading: http://www.itil-officialsite.com/home/home.asp Read: 20.10.2008

Jordan, E., Silcock, L. 2006. Strateginen IT-riskien hallinta. Edita. Helsinki.

Kostamo, E. 1999. Strateginen ajattelu Selviydy tietoyhteiskunnassa. Multiprint. Helsinki. 2000.

Kulmala, T. 2003. Ohjelmistovuokraussopimukset. Turun yliopisto oikeustieteellinen tiedekunta. Turku.

Lamberg, I. 2008. Finnish Software Business Cluster. What is Software Business? For Reading: http://www.swbusiness.fi/portal/cluster_information/__what_is_software_business_/ Read: 26.9.2008.

Laurila, J. 28.8.2007. Suomalainen ohjelmistoala kasvaa reippaasti. Talentum. Article. For Reading: http://www.talentum.com/doc.te?f_id=1215044 Read: 26.9.2008.

Laurila, J. 28.8.2007. ”Kasva lisää softafirma!”. Tietoviikko. Article. For Reading: http://www.tietoviikko.fi/kaikki_uutiset/article133326.ece?v=n Read: 24.10.2008

Lipiäinen, T. 2000. Liiketoiminta uudella vuosituhannella, liiketoiminnan suunnittelu, markkinointi ja johtaminen uudella vuosituhannella. Gummerus Kirjapaino Oy. Jyväskylä.

Lukkari, J. 12.9.2008. Ohjelmistoalan ennusteet menivät täysin metsään. Tekniikka & Talous. Article. For Reading: http://www.tekniikkatalous.fi:80/ict/article130565.ece Read: 26.10.2008

Microsoft Oy. 2008. Earning releases. FY08. For Reading: http://www.microsoft.com/msft/earnings/FY08/earn_rel_q4_08.mspx Read: 21.10.2008.

Microsoft Oy. 17.1.2006. Are you eligible for Microsoft academic discount pricing? For Reading: http://www.microsoft.com/Education/Eligible.mspx Read: 23.9.2008.

Microsoft Oy. 19.1.2004. Retail. For Reading: http://www.microsoft.com/resources/sam/partnerguide/lal_doc_retail.mspx Read: 23.9.2008.

Microsoft Oy. 2008. Volume Licensing Programs Comparison Figure. For Reading: http://download.microsoft.com/download/7/a/a/7aa89a8b-bf4d-446b-a50c-c9b00024df33/249188/VLP_Interactive_Commercial_Final_April%2007.pdf Read: 23.9.2008.

Microsoft Oy. 2008. Benefits of SAM. For Reading: http://www.microsoft.com/resources/sam/benefits.mspx Read: 19.10.2008

Mäntylä, J. 1.11.2007. DNS lähti erikoistumalla asiakkaisiin. Tietoviikko. Article. For Reading: http://www.tietoviikko.fi/taustat/kaikki_jutut/article138373.ece Read: 2.10.2008.

The National Board of Patents and Registration in Finland. 15.5.2006. General Information on patents. For Reading: http://www.prh.fi/en/patentit/hakusuomi.html Read: 5.10.2008.

The National Board of Patents and Registration in Finland. 4.12.2007. Trademarks. For Reading: http://www.prh.fi/en/tavaramerkit/tavaramerkkilyhyesti.html Read: 5.10.2008.

The National Board of Patents and Registration in Finland. 4.12.2007. Registered Designs. For Reading: http://www.prh.fi/en/mallioikeudet/mallioikeuspahkinankuoressa.html Read: 5.10.2008.

The National Board of Patents and Registration in Finland. 3.6.2008. Utility Models. For Reading: http://www.prh.fi/en/hyodyllisyysmallit.html Read: 5.10.2008.

Novell. 27.2.2007. Advanced Technical Training. Novell ZenWorks Asset management 7.5. U.S.A. 2004.

Nukari, J. Saukkonen, J. & Seppänen, V. 2003. in Ohjelmistoliiketoiminta, (toim.)Hyvönen, E. 2003. WSOY. Vantaa.

Rajala, Rossi & Tuunainen 2003. in Ohjelmistoliiketoiminta, (toim.)Hyvönen, E. 2003. WSOY. Vantaa.

Robinson, B. 2006. The next way of software licensing. Infoworld.com. 9/2006. 33 - 42.

Tietoviikko.17.3.2008. SaaS-tuotteiden käyttö yrityksissä lisääntyy vauhdilla. Article. For Reading: http://www.tietoviikko.fi/kaikki_uutiset/article137145.ece Read: 1.10.2008.

Tyrväinen, P. Lamberg, I. Nukari, J. Saukkonen, J. Seppänen, V. Warsta, J. 2005. Innovatiivisilla toimialaratkaisuilla kansainvälisille ohjelmistomarkkinoille. Technology Report 178/2005. Helsinki. For Reading:
http://www.tekes.fi/julkaisut/innovatiivisilla_toimialaratkaisuilla.pdf Read: 28.20.2008.

Tyrväinen, P, Lamberg, T, Nukari, J. Saukkonen, J. Warsta, J. 2003. in Ohjelmistoliiketoiminta, (toim.)Hyvönen, E. 2003. WSOY. Porvoo.

Ohjelmistotuoteliiketoiminnan klusterin visioprosessi. Otuli-hanke. Final report. For Reading: http://www.swbusiness.fi/uploads/attachments/1117714715_Otuli%20final%20300505.pdf Read: 1.10.2008.

Open Source Initiative. Coar, K. 7.7.2006. The Open Source Definition. For Reading: http://www.opensource.org/docs/osd Read. 15.10.2008.

Pentikäinen, J. 9.8.2007. Gartner: palvelusovellukset rymistelevät vauhdilla. Tietoviikko. Article. For Reading: http://www.tietoviikko.fi/kaikki_uutiset/article137902.ece
Read: 28.10.2008

Product Business: Results of the National Software Industry Survey 2007. Helsinki University of Technology. For Reading: http://www.sbl.tkk.fi/oskari/ Read: 27.10.2008

Rönkkö, M. Eloranta, E. Mustaniemi, H. Mutanen, O. Kontio, J. 2007. Finnish Software

Siivola, J. 2004. Immateriaalioikeudet yritysten sopimuksissa. Patentti-, hyödyllisyysmalli-, malli-, piirimalli- ja tunnusmerkkioikeudet sekä tekijänoikeudet sopimussuhteissa. Teknologiainfo Teknova Oy. Helsinki.

Sipilä, J. 1996. Asiantuntijapalveluiden tuotteistaminen. WSOY. Porvoo

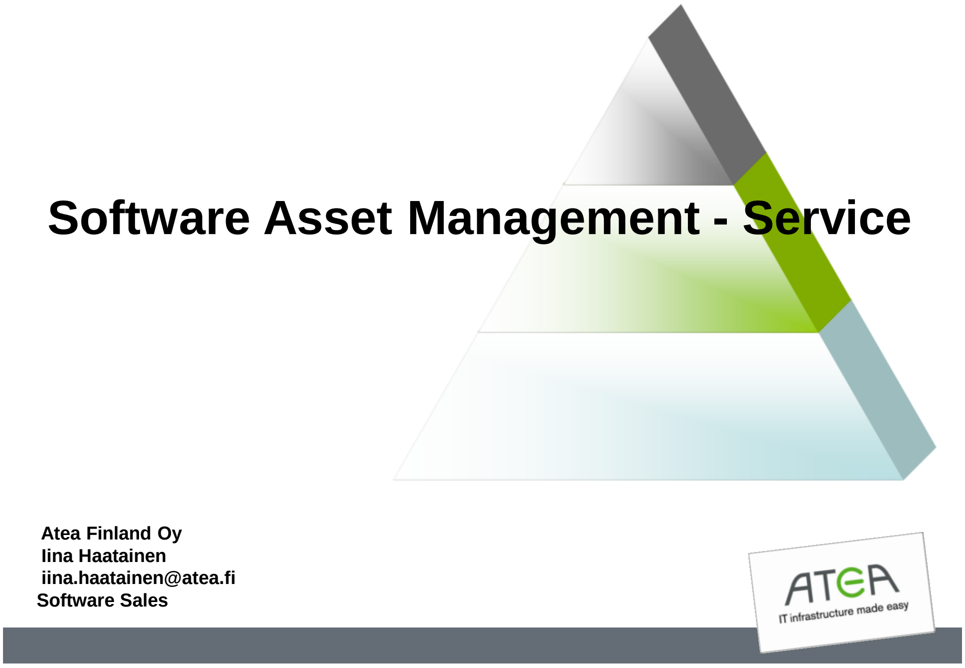
U.S Securities and Exchange Comission. 25.7.2003. Study Pursuant to Section 108(d) of the Sarbanes-Oxley Act of 2002 on the Adoption by the United States Financial Reporting System of a Principles-Based Accounting System. For Reading: http://www.sec.gov/news/studies/principlesbasedstand.htm Read: 19.10.2008

Välimäki, M. 2006. Oikeudet tietokoneohjelmistoihin ja niiden lisensiointi. Turre Bublishing. Helsinki.

Wikipedia. 2008. Freeware. For Reading: http://en.wikipedia.org/wiki/Freeware Read: 15.10.2008.

APPENDIX

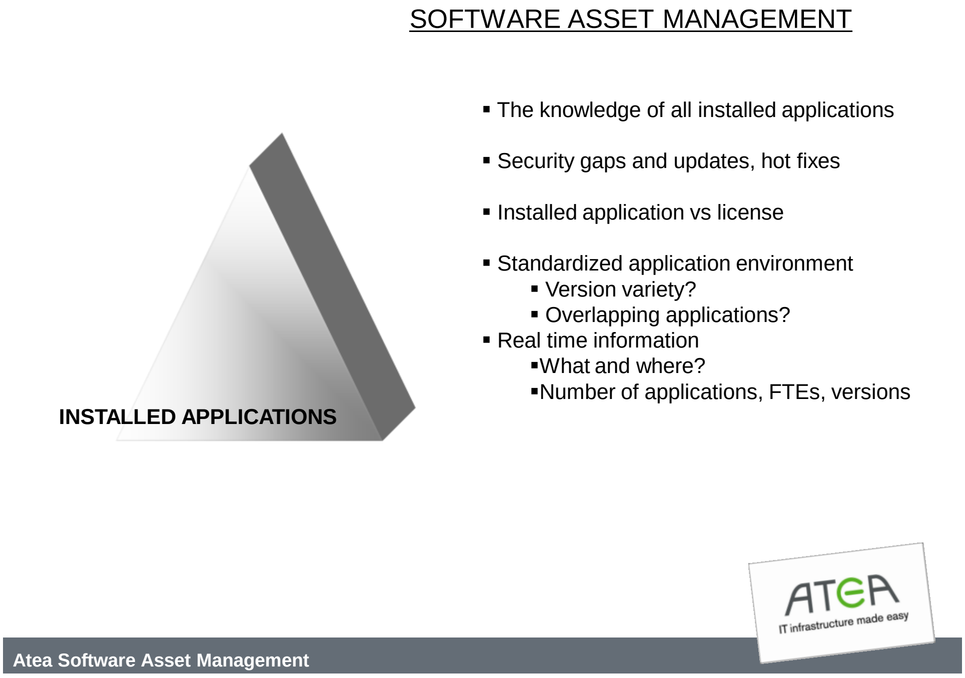
The customer presentation of the software asset management service



Software Asset Management - Service

Atea Finland Oy
Iina Haatainen
iina.haatainen@atea.fi
Software Sales

ATEA
IT infrastructure made easy

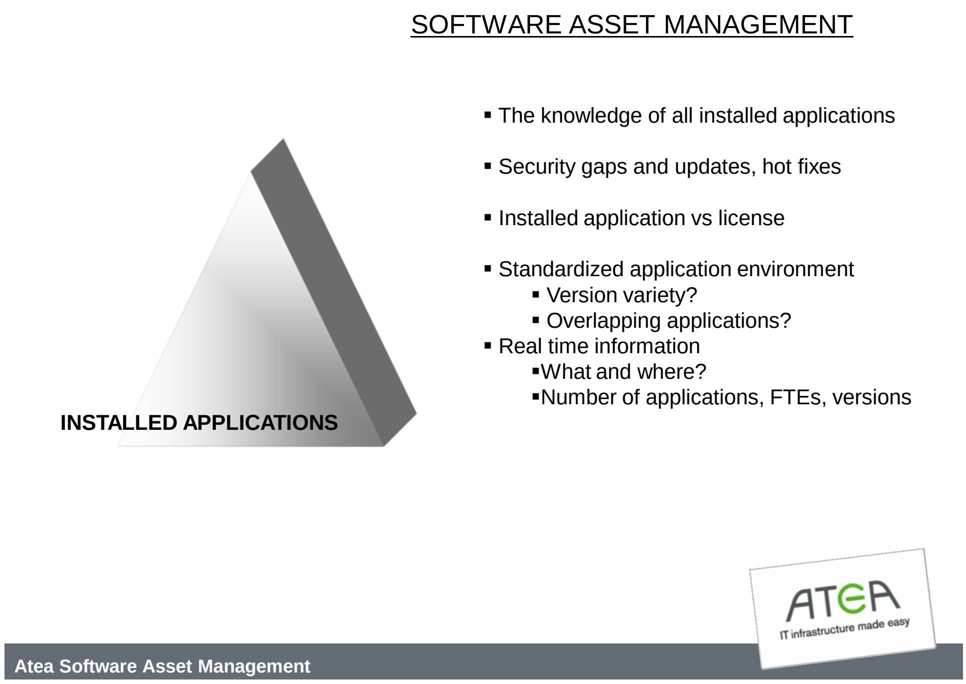


SOFTWARE ASSET MANAGEMENT

- The knowledge of all installed applications
- Security gaps and updates, hot fixes
- Installed application vs license
- Standardized application environment
  - Version variety?
  - Overlapping applications?
- Real time information
  - What and where?
  - Number of applications, FTEs, versions

INSTALLED APPLICATIONS

Atea Software Asset Management



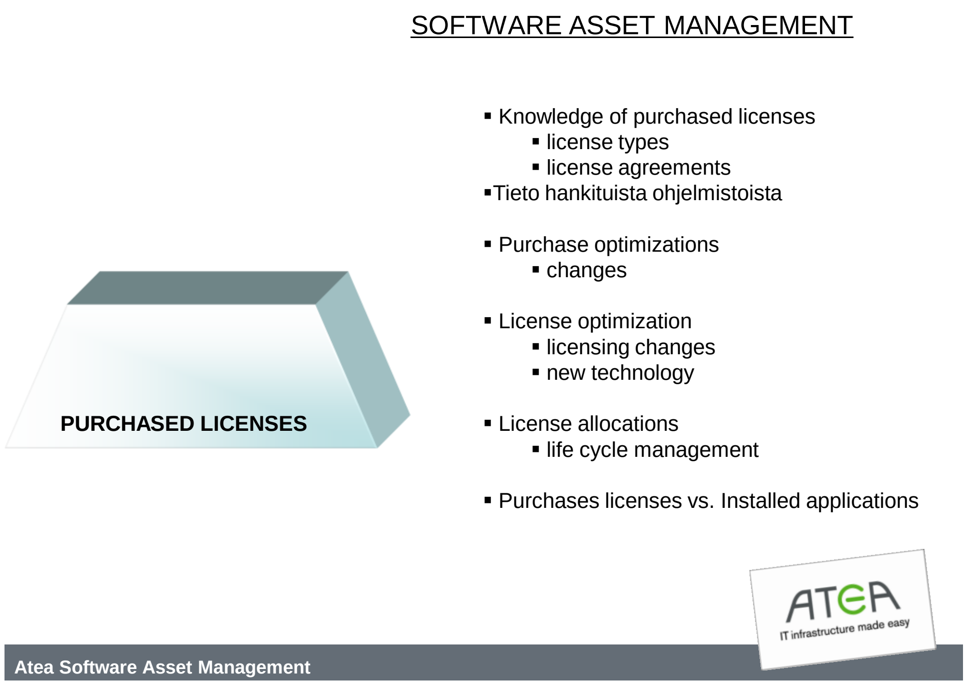
SOFTWARE ASSET MANAGEMENT

INSTALLED APPLICATIONS

- The knowledge of all installed applications
- Security gaps and updates, hot fixes
- Installed application vs license
- Standardized application environment
  - Version variety?
  - Overlapping applications?
- Real time information
  - What and where?
  - Number of applications, FTEs, versions

ATEA
IT infrastructure made easy

Atea Software Asset Management



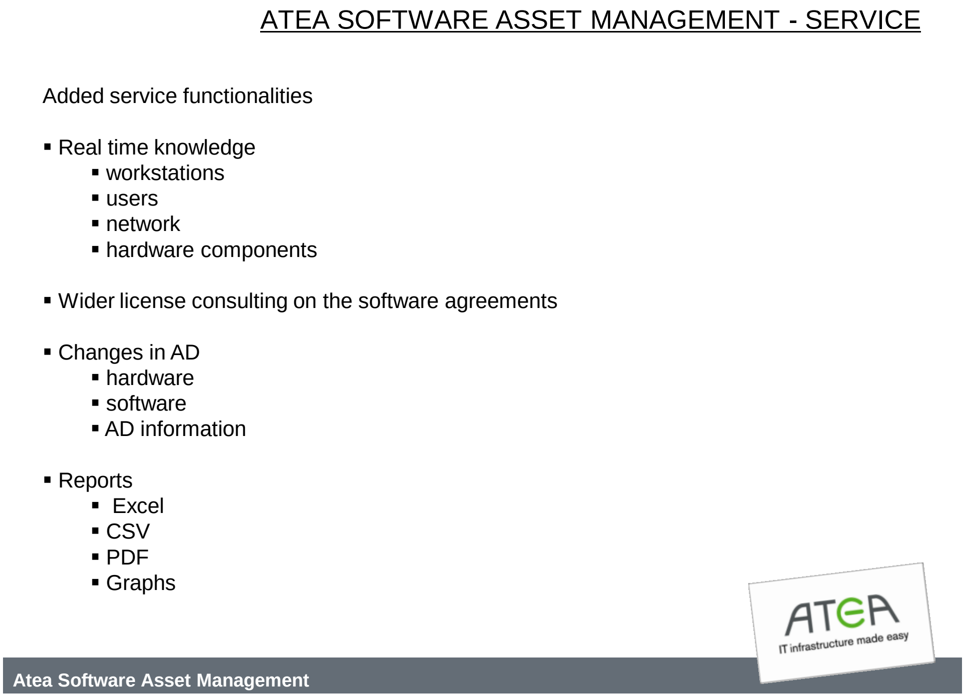
SOFTWARE ASSET MANAGEMENT

PURCHASED LICENSES

- Knowledge of purchased licenses
  - license types
  - license agreements
- Tieto hankituista ohjelmistoista
- Purchase optimizations
  - changes
- License optimization
  - licensing changes
  - new technology
- License allocations
  - life cycle management
- Purchases licenses vs. Installed applications

ATEA
IT infrastructure made easy

Atea Software Asset Management



ATEA SOFTWARE ASSET MANAGEMENT - SERVICE

Added service functionalities

- Real time knowledge
  - workstations
  - users
  - network
  - hardware components

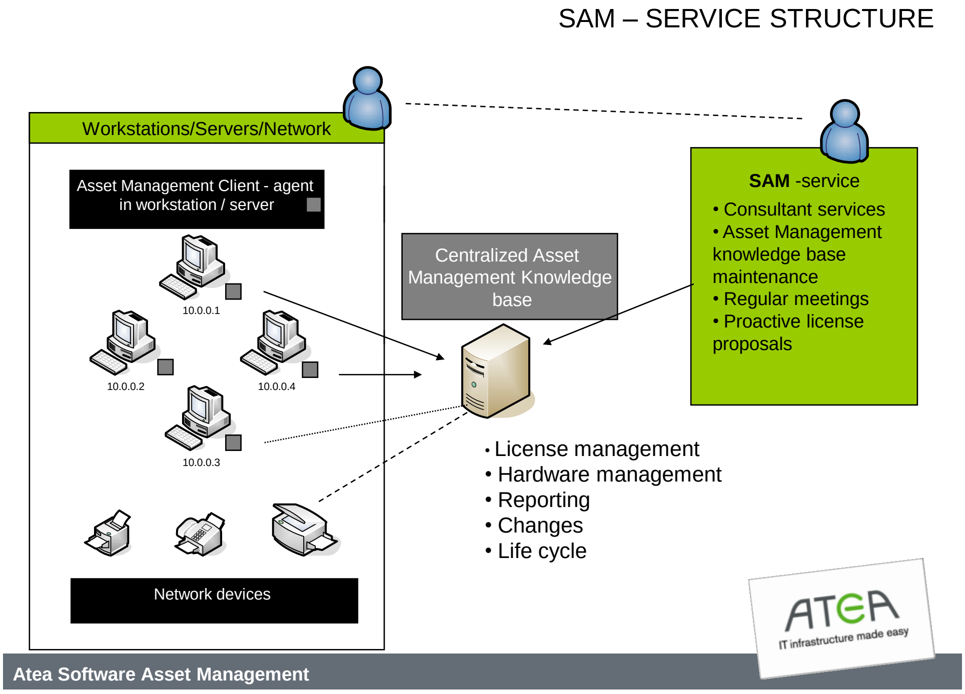
- Wider license consulting on the software agreements

- Changes in AD
  - hardware
  - software
  - AD information

- Reports
  - Excel
  - CSV
  - PDF
  - Graphs

ATEA IT infrastructure made easy

Atea Software Asset Management



SAM – SERVICE STRUCTURE

Workstations/Servers/Network

Asset Management Client - agent in workstation / server

Network devices

Centralized Asset Management Knowledge base

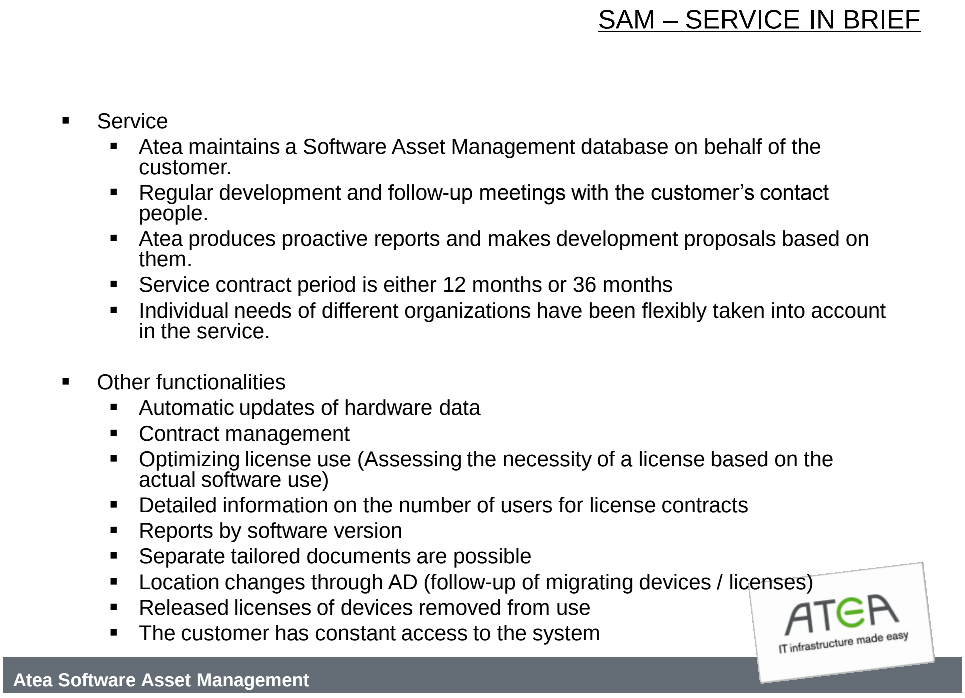
SAM -service

- Consultant services
- Asset Management knowledge base maintenance
- Regular meetings
- Proactive license proposals

- License management
- Hardware management
- Reporting
- Changes
- Life cycle

ATEA IT infrastructure made easy

Atea Software Asset Management



SAM – SERVICE IN BRIEF

- Service
  - Atea maintains a Software Asset Management database on behalf of the customer.
  - Regular development and follow-up meetings with the customer’s contact people.
  - Atea produces proactive reports and makes development proposals based on them.
  - Service contract period is either 12 months or 36 months
  - Individual needs of different organizations have been flexibly taken into account in the service.

- Other functionalities
  - Automatic updates of hardware data
  - Contract management
  - Optimizing license use (Assessing the necessity of a license based on the actual software use)
  - Detailed information on the number of users for license contracts
  - Reports by software version
  - Separate tailored documents are possible
  - Location changes through AD (follow-up of migrating devices / licenses)
  - Released licenses of devices removed from use
  - The customer has constant access to the system

Atea Software Asset Management